# Faculty of Engineering

# Savitribai Phule Pune University

# Syllabus

# of

# Second Year of Computer Engineering

# (Course 2015)

## (with effect from June 2016 )

## Prologue

It is with great pleasure and honor that I share the syllabi for Second Year of Computer Engineering (2015 Course) on behalf of Board of Studies, Computer Engineering. Savitribai Phule Pune University (SPPU) has adopted the Credit System since academic year 2015-16 and it has been successfully implemented for First year Engineering. While revising syllabus at second year with credit system, honest and sincere efforts are put to tune Computer Engineering program syllabus in tandem with the objectives of Higher Education of India, AICTE, UGC and affiliated University (SPPU) by keeping an eye on the technological advancements and industrial requirements globally.

Engineering Course sequences with core objective of topic coverage and follows the sequence as- introductory, intermediate, and advanced topics as the student progresses along the curriculum. Second year of Engineering is basically to acquaint the student with the fundamentals of the Computer Engineering stream which necessarily serves as the prerequisite for the courses to follow in third and final year.

While preparing the draft utmost care has been taken to adhere to the perspective and vision of AICTE, UGC and SPPU. This is a humble attempt of the syllabus revision which is materialized with sincere efforts, active participation, expert opinions and suggestions from domain professionals. Sincere efforts have been put by members of BOS, teachers, alumni, industry experts in framing the draft with guidelines and recommendations.

I am sincerely indebted to all the minds and hands who work adroitly to materialize the task.

**Dr. Varsha H. Patil**
**Coordinator, Board of Studies (Computer Engineering), SPPU, Pune**
**Thursday, June 9, 2016**

(This document includes- Program Educational Objectives and Program Outcomes for Computer Engineering, Second Year Courses structure for semester I and semester II, Course contents, Laboratory work guidelines, Audit Course details- options, guidelines and contents.)

# Savitribai Phule Pune University
# Computer Engineering

## Program Educational Objectives

1. To prepare globally competent graduates having strong fundamentals and domain knowledge to provide effective solutions for engineering problems.
2. To prepare the graduates to work as a committed professionals with strong professional ethics and values, sense of responsibilities, understanding of legal, safety, health, societal, cultural and environmental issues.
3. To prepare committed and motivated graduates with research attitude, lifelong learning, investigative approach, and multidisciplinary thinking.
4. To prepare the graduates with strong managerial and communication skills to work effectively as individual as well as in teams.

## Program Outcomes

**Students are expected to know and be able -**

1. To apply knowledge of mathematics, science, engineering fundamentals, problem solving skills, algorithmic analysis to solve complex engineering problems.
2. To analyze the problem by finding its domain and applying domain specific skills
3. To understand the design issues of the product/software and develop effective solutions with appropriate consideration of public health and safety, cultural, societal, and environmental issues.
4. To find solutions of complex problems by conducting investigations applying suitable techniques.
5. To adapt the usage of modern tools and recent software.
6. To contribute towards the society by understanding the impact of Engineering on global aspect.
7. To understand environment issues and design a sustainable system.
8. To understand and follow professional ethics.
9. To function effectively as an individual and as member or leader in diverse teams and interdisciplinary settings.
10. To demonstrate effective communication at various levels.
11. To apply the knowledge of Computer Engineering for development of projects, and its finance and management.
12. To keep in touch with current technologies and inculcate the practices of lifelong learning.

# Savitribai Phule Pune University

# Second Year of Computer Engineering (2015 Course)

(With effect from Academic Year 2016-17)

## Semester I

| Course Code | Course Name | Teaching Scheme Hours / Week | | | Examination Scheme & Marks | | | | | | Credit | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Theory | Tutorial | Practical | In-Sem | End-Sem | TW | PR | OR | Total | TH + TUT | PR |
| 210241 | Discrete Mathematics | 04 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 04 | -- |
| 210242 | Digital Electronics and Logic Design | 04 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 04 | -- |
| 210243 | Data Structures and Algorithms | 04 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 04 | -- |
| 210244 | Computer Organization and Architecture | 04 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 04 | -- |
| 210245 | Object Oriented Programming | 04 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 04 | -- |
| 210246 | Digital Electronics Lab | -- | -- | 02 | -- | -- | 25 | 50 | -- | 75 | -- | 01 |
| 210247 | Data Structures Lab | -- | -- | 04 | -- | -- | 25 | 50 | -- | 75 | -- | 02 |
| 210248 | Object Oriented Programming Lab | -- | -- | 02 | -- | -- | 25 | 50 | -- | 75 | -- | 01 |
| 210249 | Soft Skills | -- | -- | 02 | -- | -- | 25 | -- | -- | 25 | -- | 01 |
| Total | | | | | | | | | | | 20 | 05 |
| 210250 | Audit Course 1 | -- | -- | -- | -- | -- | -- | -- | -- | -- | Grade | |
| Total | | 20 | -- | 10 | 250 | 250 | 100 | 150 | -- | 750 | 25 | |

## Abbreviations:

| | | | |
|---|---|---|---|
| TW: | Term Work | TH: | Theory |
| OR: | Oral | TUT: | Tutorial |
| PR: | Practical | Sem: | Semester |

## Savitribai Phule Pune University
## Second Year of Computer Engineering (2015 Course)
### (With effect from Academic Year 2016-17)
## Semester II

| Course Code | Course Name | Teaching Scheme Hours / Week | | | Examination Scheme & Marks | | | | | | Credits | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Theory | Tutorial | Practical | In-Sem | End-Sem | TW | PR | OR | Total | TH+ TUT | PR |
| 207003 | Engineering Mathematics III | 04 | 01 | -- | 50 | 50 | 25 | -- | -- | 125 | 05 | -- |
| 210251 | Computer Graphics | 04 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 04 | -- |
| 210252 | Advanced Data Structures | 04 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 04 | -- |
| 210253 | Microprocessor | 04 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 04 | -- |
| 210254 | Principles of Programming Languages | 03 | -- | -- | 50 | 50 | -- | -- | -- | 100 | 03 | -- |
| 210255 | Computer Graphics Lab | -- | -- | 02 | -- | -- | 25 | 50 | -- | 75 | -- | 01 |
| 210256 | Advanced Data Structures Lab | -- | -- | 04 | -- | -- | 25 | 50 | -- | 75 | -- | 02 |
| 210257 | Microprocessor Lab | -- | -- | 04 | -- | -- | 25 | 50 | -- | 75 | -- | 02 |
| Total | | | | | | | | | | | 20 | 05 |
| 210258 | Audit Course 2 | | -- | -- | -- | -- | -- | -- | -- | -- | Grade | |
| Total | | 19 | 01 | 10 | 250 | 250 | 100 | 150 | -- | 750 | 25 | |

### Abbreviations:

| | | | |
|---|---|---|---|
| TW: | Term Work | TH: | Theory |
| OR: | Oral | TUT: | Tutorial |
| PR: | Practical | Sem: | Semester |

# SEMESTER I

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210241: Discrete Mathematics

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | 04 | In-Sem (online): 50 Marks<br>End-Sem (paper): 50 Marks |

**Prerequisite:-** Basic Mathematics

**Course Objectives:**

- To use appropriate set, function and relation models to understand practical examples, and interpret the associated operations and terminologies in context.
- Determine number of logical possibilities of events.
- Learn logic and proof techniques to expand mathematical maturity.
- Formulate problems precisely, solve the problems, apply formal proof techniques, and explain the reasoning clearly.

**Course Outcomes:**

On completion of the course, student will be able to–

- Solve real world problems logically using appropriate set, function, and relation models and interpret the associated operations and terminologies in context.
- Analyze and synthesize the real world problems using discrete mathematics.

## Course Contents

| Unit I | Set Theory and Logic | 09 Hours |
|---|---|---|

Discrete Mathematics, Significance of Discrete Mathematics in Computer Engineering, **Sets**– Naïve Set Theory (Cantorian Set Theory), Axiomatic Set Theory, Need for Sets, Representation of Sets, Set Operations, cardinality of set, principle of inclusion and exclusion, **Types of Sets** – Countable and Uncountable Sets, Finite and Infinite Sets, Countably Infinite and Uncountably Infinite Sets. Introduction to bounded and unbounded sets and multiset. Countability of Rational Numbers Using Cantor Diagonalization Argument, power set. **Propositional Logic-** logic**,** Propositional Equivalences, Application of Propositional Logic-Translating English Sentences, Proof by Mathematical Induction and Strong Mathematical Induction.

| Unit II | Relations and Functions | 09 Hours |
|---|---|---|

Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations , Closures of Relations, Equivalence Relations, Partial Orderings, partitions, Hasse Diagram, Lattices, Chains and Anti-Chains, Transitive Closure and Warshall's Algorithm, n-Ary Relations and their Applications.

**Functions-** Surjective, Injective and Bijective functions, Inverse Functions and Compositions of Functions, The Pigeonhole Principle.

| Unit III | Counting | 09 Hours |
|---|---|---|
| The Basics of Counting, rule of Sum and Product, Permutations and Combinations, Binomial Coefficients and Identities, Generalized Permutations and Combinations, Algorithms for generating Permutations and Combinations. | | |
| **Unit IV** | **Graph Theory** | **09 Hours** |
| Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Single source shortest path- Dijkstra's Algorithm, Planar Graphs, Graph Colouring. Case Study- Web Graph, Google map. | | |
| **Unit V** | **Trees** | **09 Hours** |
| Introduction, properties of trees, Binary search tree, decision tree, prefix codes and Huffman coding, cut sets, Spanning Trees and Minimum Spanning Tree, Kruskal's and Prim's algorithms, The Max flow- Min Cut Theorem (Transport network). Case Study- Game Tree, Mini-Max Tree. | | |
| **Unit VI** | **Algebraic Structures and Coding Theory** | **09 Hours** |
| The structure of algebra, Algebraic Systems, Semi Groups, Monoids, Groups, Homomorphism and Normal Subgroups, and congruence relations, Rings, Integral Domains and Fields, coding theory, Polynomial Rings and polynomial Codes, **Case Study-** Brief introduction to Galois Theory –Field Theory and Group Theory. | | |

**Books:**

**Text:**

1. Kenneth H. Rosen, "Discrete Mathematics and its Applications", Tata McGraw-Hill, ISBN 978-0-07-288008-3, 7th Edition.
2. C. L. Liu, "Elements of Discrete Mathematics", TMH, ISBN 10:0-07-066913-9.

**References:**

1. Bernard Kolman, Robert C. Busby and Sharon Ross, "Discrete Mathematical Structures", Prentice-Hall of India /Pearson, ISBN: 0132078457, 9780132078450.
2. N. Biggs, "Discrete Mathematics", 3rd Edition, Oxford University Press, ISBN 0 –19 850717 – 8.
3. Narsingh Deo, "Graph with application to Engineering and Computer Science", Prentice Hall of India, 1990, 0 – 87692 – 145 – 4.
4. Dr. K. D. Joshi, " Foundations of Discrete Mathematics", New Age International Limited, Publishers, January 1996, ISBN: 8122408265, 9788122408263
5. C.D. Cantrell, "Modern Mathematical Methods for Engineers", Cambridge University Press, ISBN-0521670497
6. Eric Gossett, "Discrete Mathematical Structures with Proofs", Wiley India Ltd, ISBN:978-81-265-2758-8.
7. Sriram P & Steven S, "Computational Discrete Mathematics", Cambridge University Press, ISBN 13: 978-0-521-73311-3.

# Savitribai Phule Pune University
## Second Year of Computer Engineering (2015 Course)
## 210242: Digital Electronics & Logic Design

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | 04 | In-Sem (online): 50 Marks<br>End-Sem (paper): 50 Marks |

**Prerequisite: -** Basic Electronics Engineering

### Course Objectives:

- To understand the functionality and design of Combinational and Sequential Circuits
- To understand and compare the functionalities, properties and applicability of Logic Families.
- To understand concept of programmable logic devices and ASM chart and get acquainted with design of synchronous state machines.
- To design and implement digital circuits using VHDL.

### Course Outcomes:

On completion of the course, student will be able to–

- Realize and simplify Boolean Algebraic assignments for designing digital circuits using K-Maps.
- Design and implement Sequential and Combinational digital circuits as per the specifications.
- Apply the knowledge to appropriate IC as per the design specifications.
- Design simple digital systems using VHDL.
- Develop simple embedded system for simple real world application.

## Course Contents

| Unit I | Combinational Logic Design | 09 Hours |
|---|---|---|

**Logic minimization:** Representation of truth-table, Sum of Product (SOP) form, Product of Sum (POS) form, Simplification of logical functions, Minimization of SOP and POS forms using K-Maps up to 4 variables and Quine-McCluskey Technique, realization of logic gates.

**Design of Combinational Logic:** Code converter **-** BCD, Excess-3, Gray code, Binary Code. Half- Adder, Full Adder, Half Subtractor, Full Subtractor, Binary Adder (IC 7483), BCD adder, Look ahead carry generator, Multiplexers (MUX): MUX (IC 74153, 74151), MUX tree, Demultiplexers (DEMUX)**-** Decoder. (IC 74138, IC 74154). DMUX Tree, Implementation of SOP and POS using MUX, DMUX, Comparators, Parity generators and Checker, Priority Encoders.

| Unit II | Sequential Logic Design | 09 Hours |
|---|---|---|

**Flip- flop:** SR, JK, D, T; Preset & Clear, Master and Slave Flip Flops, Truth Tables and Excitation tables, Conversion from one type to another type of Flip Flop. **Registers:** Buffer register, shift register, Applications of shift registers. **Counters:** Asynchronous counter. Synchronous counter, ring counters, BCD Counter, Johnson Counter, Modulus of the counter (IC 7490).

**Synchronous Sequential Circuit Design:** Models – Moore and Mealy, State diagram and State Tables, Design Procedure, Sequence generator and detector. **Asynchronous Sequential Circuit Design:** Difference with synchronous circuit design, design principles and procedure, applications.

### Unit III Algorithmic State Machines 09 Hours

**Algorithmic State Machines:** Finite State Machines (FSM) and ASM, ASM charts, notations, construction of ASM chart and realization for sequential circuits, Sequence Generator, Types of Counters. **VHDL:** Introduction to HDL, Data Objects & Data Types, Attributes., VHDL- Library, Design Entity, Architecture, Modeling Styles, Concurrent and Sequential Statements,

**Design Examples:** VHDL for Combinational Circuits-Adder, MUX, VHDL for Sequential Circuits, Synchronous and Asynchronous Counter.

### Unit IV Programmable Logic Devices 09 Hours

ROM as PLD, Programmable Logic Array (PLA), Programmable Array Logic (PAL), Designing combinational circuits using PLDs.

### Unit V Logic Families 09 Hours

**Classification of logic families:** Unipolar and Bipolar Logic Families, Characteristics of Digital ICs: Speed, power dissipation, figure of merits, fan-out, Current and voltage parameters, Noise immunity, operating temperature range, power supply requirements. **Transistor-Transistor Logic:** Operation of TTL, Current sink logic, TTL with active pull up, TTL with open collector output, Schottkey TTL, TTL characteristics, TTL 5400/7400 series, **CMOS:** CMOS Inverter, CMOS characteristics, CMOS configurations- Wired Logic, Open drain outputs, **Interfacing:** TTL to CMOS and CMOS to TTL. Tristate Logic and Tristate TTL inverter.

### Unit VI Microcontrollers 09 Hours

Comparison of typical microprocessor and microcontroller. **Microcontroller 8051:** Features, architecture, Pin description, **Programming model**– Special Function Registers, addressing modes, instruction set, Timers and Counters, serial communication, interrupts, interfacing with ADC and DAC.

### Books:

#### Text:

1. R.P. Jain, “Modern Digital Electronics”, TMH, 2012, ISBN–13: 978-0-07- 066911-6.
2. Stephen Brown, Zvonko Vranesic, “Fundamentals of Digital Logic with VHDL Design”, McGraw-Hill, ISBN–13:978-1-25-902597-6.
3. Muhammas Mazidi, Janice Mazidi and Rolin McKinlay, “The 8051 Microcontroller and Embedded Systems using Assembly and C”, Pearson Education, ISBN-13: 9788131758991

#### References:

1. John Yarbrough, “Digital Logic applications and Design”, Cengage Learning, ISBN – 13: 978-81-315-0058-3
2. D. Leach, Malvino, Saha, “Digital Principles and Applications”, Tata McGraw Hill, ISBN – 13:978-0-07-014170-4.
3. Anil Maini, “Digital Electronics: Principles and Integrated Circuits”, Wiley India Ltd, ISBN:978-81-265-1466-3.
4. Norman B & Bradley, “Digital Logic Design Principles, Wiley India Ltd, ISBN:978-81-265-1258-4.
5. Scott Mackenzie, “The 8051 Microcontroller”, Prentice Hall India, ISBN-13: 978-0130195623

# Savitribai Phule Pune University
## Second Year of Computer Engineering (2015 Course)
## 210243: Data Structures and Algorithms

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | 04 | In-Sem (online): 50 Marks<br>End-Sem (paper): 50 Marks |

**Prerequisites: -** FPL I and FPL II

**Course Objectives:**

- To understand the standard and abstract data representation methods.
- To acquaint with the structural constraints and advantages in usage of the data.
- To understand the memory requirement for various data structures.
- To operate on the various structured data.
- To understand various data searching and sorting methods with pros and cons.
- To understand various algorithmic strategies to approach the problem solution.

**Course Outcomes:**

On completion of the course, student will be able to–

- To discriminate the usage of various structures in approaching the problem solution.
- To design the algorithms to solve the programming problems.
- To use effective and efficient data structures in solving various Computer Engineering domain problems.
- To analyze the problems to apply suitable algorithm and data structure.
- To use appropriate algorithmic strategy for better efficiency

## Course Contents

| | | |
|---|---|---|
| Unit I | Introduction to Algorithm and Data Structures | 09 Hours |

**Algorithms-** Problem Solving, Introduction to Algorithms, Characteristics of algorithms, Algorithm design tools: Pseudo code and flowchart, Analysis of Algorithms, Complexity of algorithms- Space complexity, Time complexity, Asymptotic notation- Big-O, Theta and Omega, standard measures of efficiency.

**Data Structures-** Data structure, Abstract Data Types (ADT), Concept of linear and Non-linear, static and dynamic, persistent and ephemeral data structures, and relationship among data, data structure, and algorithm, From Problem to Program.

**Algorithmic Strategies-** Introduction to algorithm design strategies- Divide and Conquer, and Greedy strategy.

**Recurrence relation -** Recurrence Relation, Linear Recurrence Relations, With constant Coefficients, Homogeneous Solutions. Solving recurrence relations

| | | |
|---|---|---|
| Unit II | Linear Data Structures Using Sequential Organization | 09 Hours |

Sequential Organization, Linear Data Structure Using Sequential Organization, Array as an Abstract Data Type, Memory Representation and Address Calculation, Inserting an element into an array, Deleting an element, Multidimensional Arrays, Two-dimensional arrays, n- dimensional arrays, Concept of Ordered List, Single Variable Polynomial, Representation using arrays, Polynomial as array of structure, Polynomial addition, Polynomial multiplication, Sparse Matrix, Sparse matrix representation, Sparse matrix addition, Transpose of sparse matrix, String Manipulation Using Array. **Case Study-** Use of sparse matrix in Social Networks and Maps.

| | | |
|---|---|---|
| Unit III | Linked Lists | 09 Hours |

Concept, Comparison of sequential and linked organizations, Primitive operations, Realization of Linked Lists, Realization of linked list using arrays, Dynamic Memory Management, Linked list using dynamic memory management, Linked List Abstract Data Type, Linked list operations, Head pointer and header node, **Types of linked list-** Linear and circular linked lists, Doubly Linked List and operations, Circular Linked List, Singly circular linked list, Doubly circular linked list**, Polynomial Manipulations -** Polynomial addition, Multiplication of two polynomials using linked list. **Generalized Linked List (GLL)** concept, representation of polynomial and sets using GLL. **Case Study-** Garbage Collection.

### Unit IV Stacks 09 Hours

**Stacks-** concept, Primitive operations, Stack Abstract Data Type, Representation of Stacks Using Sequential Organization, stack operations, Multiple Stacks, Applications of Stack- Expression Evaluation and Conversion, Polish notation and expression conversion, Need for prefix and postfix expressions, Postfix expression evaluation, Linked Stack and Operations**.**

**Recursion-** concept, **variants of recursion-** direct, indirect, tail and tree, Backtracking algorithmic strategy, use of stack in backtracking. **Case Study-** 4 Queens problem, Android-multiple tasks/multiple activities and back stack.

### Unit V Queues 09 Hours

Concept, Queue as Abstract Data Type, Realization of Queues Using Arrays , Circular Queue, Advantages of using circular queues, Multi-queues, Deque, Priority Queue, Array implementation of priority queue, Linked Queue and operations. **Case study- P**riority queue in bandwidth management.

### Unit VI Sorting and Searching 09 Hours

**Searching-** Search Techniques, Sequential search, variant of sequential search- sentinel search, Binary search, Fibonacci search. **Case Study**- Use of Fibonacci search in non-uniform access memory storage and in Optimization of Unimodal Functions. **Sorting-** Types of sorting-Internal and external sorting, General sort concepts-sort order, stability, efficiency, number of passes, Sorting methods- Bubble sort, Insertion sort, Selection sort, Quick sort, Heap sort, Shell sort, Bucket sort, Radix sort, Comparison of All Sorting Methods. **Case Study-** Timsort as a hybrid stable sorting algorithm.

### Books:

**Text:**

1. Brassard & Bratley, "Fundamentals of Algorithmics", Prentice Hall India/Pearson Education, ISBN 13-9788120311312**.**
2. Horowitz and Sahani, "Fundamentals of Data Structures in C++", University Press, ISBN 10: 0716782928 ISBN 13: 9780716782926.
3. Goodrich, Tamassia, Goldwasser, "Data Structures and Algorithms in C++", Wiley publication, ISBN-978-81-265-1260-7

**References:**

1. R. Gillberg, B. Forouzn, "Data Structures: A Pseudo code approach with C", Cenage Learning, ISBN 9788131503140.
2. Horowitz, Sahani and Rajshekaran, "Fundamentals of Computer Algorithms", University Press, ISBN-13, 9788175152571.
3. Yedidyah Langsam, Moshe J Augenstein, Aron M Tenenbaum, "Data Structures using C and C++", Pearson Education, ISBN 81-317-0328-2.
4. A Michael Berman, "Data Structures via C++: Objects by Evolution", Oxford University Press, ISBN:0-19-510843-4.
5. M. Weiss, "Data Structures and Algorithm Analysis in C++", 2nd edition, Pearson Education, 2002, ISBN-81-7808-670-0.

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210244: Computer Organization and Architecture

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | 04 | In-Sem (online): 50 Marks<br>End-Sem (paper): 50 Marks |

**Prerequisites: -**

- Fundamentals of Programming Languages-I & II and
- Basics of Electronics Engineering

**Course Objectives:**

- To understand the structure, function and characteristics of computer systems.
- To understand the design of the various functional units and components of digital computers.
- To identify the elements of modern instructions sets and explain their impact on processor design.
- To explain the function of each element of a memory hierarchy, identify and compare different methods for computer I/O.
- To compare simple computer architectures and organizations based on established performance metrics.

**Course Outcomes:**

On completion of the course, student will be able to–

- Demonstrate computer architecture concepts related to design of modern processors, memories and I/Os.
- Analyze the principles of computer architecture using examples drawn from commercially available computers.
- Evaluate various design alternatives in processor organization.

## Course Contents

| Unit I | Computer Evolution and Performance | 09 Hours |
|---|---|---|

Computer Organization and Architecture, Structure and Function, Evolution (a brief history) of computers, Designing for Performance, Evolution of Intel processor architecture- 4 bit to 64 bit, performance assessment. **A top level view of Computer function and interconnection-** Computer Components, Computer Function, Interconnection structure, bus interconnection, **Computer Arithmetic-** The Arithmetic and Logic Unit, addition and subtraction of signed numbers, design of adder and fast adder, carry look ahead addition, multiplication of positive numbers, signed operand multiplication, booths algorithm, fast multiplication, integer division. **Floating point representation and operations** – IEEE standard, arithmetic operations, guard bits and truncation.

| Unit II | Computer Memory System | 09 Hours |
|---|---|---|

Characteristics of memory system, The memory hierarchy. **Cache Memory-** Cache memory principles, Elements of cache design- cache address, size, mapping functions, replacement algorithms, write policy, line size, number of cache, one level and two level cache, performance characteristics of two level cache- locality & operations. **Case Study-** PentiumIV cache organization. **Internal Memory-** semiconductor main memory, advanced DRAM organization. **External Memory-** Hard Disk organization, RAID- level 1 to level 6.

### Unit III Input and Output System 09 Hours

External devices, **I/O modules**- Module function and I/O module structure, **Programmed I/O**-overview, I/O commands, I/O instructions, Interrupt driven I/O- interrupt processing, design issues. **Case Study-** Study of Programmable Interrupt Controller Intel 82C59A in brief. **Direct Memory Access-** drawbacks of programmed and interrupt driven I/O, DMA functions, **Case Study- DMA Controller** Intel 8237A-study in brief, I/O channels and processors- evolution and characteristics, The external Interface- Thunderbolt and Infinite Band.

### Unit IV Instruction Sets 09 Hours

**Characteristics and Functions-** machine instruction characteristics, types of operands, **Case Study-**Intel 8086, Types **of operations-** data transfer, arithmetic, logical, conversion, input-output, system control, and transfer of control, **Case Study-**Intel 8086 operation types. **Addressing modes and Formats-** Addressing modes- immediate, direct, indirect, register, register indirect, displacement and stack, **Case Study-**8086 addressing modes, **Instruction Formats-** instruction length, allocation of bits, variable length instructions. **Case Study-** 8086 instruction formats.

### Unit V Processor Organization 09 Hours

Processor organization, Register organization- user visible registers, control and status registers, **Case Study-** register organization of microprocessor 8086. **Instruction Cycle-** The machine cycle and Data flow. **Instruction Pipelining-** Pipelining Strategy, pipeline performance, pipeline hazards, dealing with branches, **Case Study**- pipelining in Pentium.

**Instruction level parallelism and superscalar processors-** Super scalar verses super pipelined, constraints, **Design Issues-** instruction level and machine parallelism, Instruction issue policy, register renaming, machine parallelism, branch prediction, superscalar execution and implementation. **Case study-** Pentium IV.

### Unit VI Basic Processing Unit 09 Hours

**Fundamental Concepts-** register transfer, performing arithmetic or logic operations, fetching a word from memory, storing a word in memory, Execution of a complete instruction- branch instructions,

**Hardwired control**, M**icro-programmed control-** micro instructions, micro program sequencing, wide branch addressing, microinstruction with next address field, pre-fetching microinstructions and emulation.

### Books:

**Text:**

1. W. Stallings, “Computer Organization and Architecture: Designing for performance”, Pearson Education/ Prentice Hall of India, 2003, ISBN 978-93-325-1870-4, 7th Edition.
2. Zaky S, Hamacher, “Computer Organization”, 5th Edition, McGraw-Hill Publications, 2001, ISBN- 978-1-25-900537-5, 5th Edition.

### References:

1. John P Hays, “Computer Architecture and Organization”, McGraw-Hill Publication, 1998, ISBN:978-1-25-902856-4, 3rd Edition.
2. Miles Murdocca and Vincent Heuring, “Computer Architecture and Organization- an integrated approach, Wiley India Pvt. Ltd, ISBN:978-81-265-1198-3, 2nd Edition
3. A. Tanenbaum, “Structured Computer Organization”, Prentice Hall of India, 1991 ISBN: 81 – 203 – 1553 – 7, 4th Edition
4. Patterson and Hennessy, “Computer Organization and Design”, Morgan Kaufmann Publishers In, ISBN 978-0-12-374750-1, 4th Edition.

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210245: Object Oriented Programming

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | 04 | In-Sem (online): 50 Marks<br>End-Sem (paper): 50 Marks |

**Prerequisites:** Fundamentals of Programming Languages-I and II

## Course Objectives:

- To explore the principles of Object Oriented Programming (OOP).
- To understand object-oriented concepts such as data abstraction, encapsulation, inheritance, dynamic binding, and polymorphism.
- To use the object-oriented paradigm in program design.
- To lay a foundation for advanced programming.
- Provide programming insight using OOP constructs.

## Course Outcomes:

On completion of the course, student will be able to–

- Analyze the strengths of object oriented programming
- Design and apply OOP principles for effective programming
- Develop programming application using object oriented programming language C++
- Percept the utility and applicability of OOP

## Course Contents

### Unit I — Classes and Objects — 09 Hours

Need of Object-Oriented Programming (OOP), Object Oriented Programming Paradigm, Basic Concepts of Object-Oriented Programming, Benefits of OOP, C++ as object oriented programming language.

**C++ Programming**- C++ programming Basics, Data Types, Structures, Enumerations, control structures, Arrays and Strings, Class, Object, class and data abstraction, class scope and accessing class members, separating interface from implementation, controlling access to members.

**Functions-** Function, function prototype, accessing function and utility function, Constructors and destructors, Copy Constructor, Objects and Memory requirements, Static Class members, data abstraction and information hiding, inline function.

### Unit II — Polymorphism and Inheritance — 09 Hours

**Operator Overloading-** concept of overloading, operator overloading, Overloading Unary Operators, Overloading Binary Operators, Data Conversion, Type casting (implicit and explicit), Pitfalls of Operator Overloading and Conversion, Keywords explicit and mutable.

**Inheritance-** Base Class and derived Class, protected members, relationship between base Class and derived Class, Constructor and destructor in Derived Class, Overriding Member Functions, Class Hierarchies, Inheritance, Public and Private Inheritance, Levels of Inheritance, Multiple Inheritance, Ambiguity in Multiple Inheritance, Aggregation, Classes Within Classes.

**Polymorphism-** concept, relationship among objects in inheritance hierarchy, abstract classes, polymorphism.

| Unit III | Virtual Functions | 09 Hours |
|---|---|---|

**Virtual Functions-** Pointers- indirection Operators, Memory Management: new and delete, Pointers to Objects, A Linked List Example, accessing Arrays using pointers, Function pointers, Pointers to Pointers, A Parsing Example, Debugging Pointers, Dynamic Pointers, smart pointers, shared pointers, Case Study : Design of Horse Race Simulation.

**Virtual Function-** Friend Functions, Static Functions, Assignment and Copy Initialization, this Pointer, virtual function, dynamic binding, Virtual destructor.

| Unit IV | Templates and Exception Handling | 09 Hours |
|---|---|---|

**Templates-** function templates, Function overloading, overloading Function templates, class templates, class template and Nontype parameters, template and inheritance, template and friends Generic Functions, Applying Generic Function, Generic Classes, The typename and export keywords, The Power of Templates.

**Exception Handling-** Fundamentals, other error handling techniques, simple exception handling-Divide by Zero, rethrowing an exception, exception specifications, processing unexpected exceptions, stack unwinding, constructor, destructor and exception handling, exception and inheritance.

| Unit V | Files and Streams | 09 Hours |
|---|---|---|

Data hierarchy, Stream and files, Stream Classes, Stream Errors, Disk File I/O with Streams, File Pointers, and Error Handling in File I/O, File I/O with Member Functions, Overloading the Extraction and Insertion Operators, memory as a Stream Object, Command-Line Arguments, Printer output, Early vs. Late Binding.

| Unit VI | Standard Template Library (STL) | 09 Hours |
|---|---|---|

Introduction to STL, Containers, algorithms and iterators, Containers- Sequence container and associative containers, container adapters, Algorithms- basic searching and sorting algorithms, min-max algorithm, set operations, heap sort, Iterators- input, output, forward, bidirectional and random access. Object Oriented Programming – a road map to future

## Books:

### Text:

1. Bjarne Stroustrup, “The C++ Programming language”, Third edition, Pearson Education. ISBN 9780201889543.
2. **Deitel,** “C++ How to Program”, 4th Edition, Pearson Education, ISBN:81-297-0276-2

### References:

1. Robert Lafore, “Object-Oriented Programming in C++”, fourth edition, Sams Publishing, ISBN:0672323087 (ISBN 13: 9780672323089)
2. Herbert Schildt, “C++ The complete reference”, Eighth Edition, McGraw Hill Professional, 2011, ISBN:978-00-72226805
3. Matt Weisfeld, “The Object-Oriented Thought Process”, Third Edition Pearson ISBN-13:075-2063330166
4. Cox Brad, Andrew J. Novobilski, “Object –Oriented Programming: An Evolutionary Approach”, Second Edition, Addison–Wesley, ISBN:13:978-020-1548341

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210246: Digital Electronics Lab

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| PR: 02 Hours/Week | 01 | TW: 25 Marks<br>PR: 50 Marks |

## Guidelines for Instructor's Manual

The instructor’s manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface etc), University syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, data sheets of various ICs, 8051 simulator and references.

## Guidelines for Student's Lab Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept, circuit diagram, pin configuration, conclusion/analysis).

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided.

## Guidelines for Lab /TW Assessment

Continuous assessment of laboratory work is done based on overall performance and lab performance of student. Each lab assignment assessment should assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficiency, punctuality and neatness.

## Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. Student should perform at least 14 experiments-5 experiments from group A and 5 assignments from group B, 2 from group C and 2 from group D.

## Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student’s for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

| Suggested List of Laboratory Assignments | |
|---|---|
| **Sr No** | **Group A** |
| 1. | Realize Full Adder and Subtractor using a) Basic Gates and b) Universal Gates |
| 2. | Design and implement Code converters-Binary to Gray and BCD to Excess-3 |
| 3. | Design of n-bit Carry Save Adder (CSA) and Carry Propagation Adder (CPA). Design and Realization of BCD Adder using 4-bit Binary Adder (IC 7483). |
| 4. | Realization of Boolean Expression for suitable combination logic using MUX 74151 / DMUX 74154 |
| 5. | Verify the truth table of one bit and two bit comparators using logic gates and comparator IC |
| 6. | Design & Implement Parity Generator using EX-OR. |
| | **Group B** |
| 7. | Flip Flop Conversion: Design and Realization |
| 8. | Design and implement a system using flip-flops, to monitor number of vehicles entering and exiting from a car parking area with maximum capacity of 15 and having separate entry and exit gates. |
| 9. | Design of Ripple Counter using suitable Flip Flops |
| 10. | a. Realization of 3 bit Up/Down Counter using MS JK Flip Flop / D Flip Flop<br>b. Realization of Mod -N counter using ( 7490 and 74193 ) |
| 11. | Assume a scenario of a hall where students are entering to attend seminar. Design and implement a system which will increment count if student is entering in the hall and will decrement count if student is exiting the hall. Assume seating capacity of a hall is 63. |
| 12. | Design and Realization of Ring Counter and Johnson Ring counter. |
| 13. | Design and implement Sequence generator using JK flip-flop. |
| 14. | Design and implement pseudo random sequence generator. |
| 15. | Design and implement Sequence detector using JK flip-flop |
| 16. | Design of ASM chart using MUX controller Method. |
| | **Group C** |
| 17. | Design and Implementation of Combinational Logic using PLAs. |
| 18. | Design and simulation of - Full adder , Flip flop, MUX using VHDL (Any 2)<br>Use different modeling styles. |
| 19. | Design & simulate asynchronous 3- bit counter using VHDL. |
| 20. | Design and Implementation of Combinational Logic using PALs. |
| **Group D (Study Assignments)** | |
| 21. | Study of Shift Registers ( SISO,SIPO, PISO,PIPO ) |
| 22. | Study of TTL Logic Family: Feature, Characteristics and Comparison with CMOS Family |
| 23. | Study of Microcontroller 8051 : Features, Architecture and Programming Model |

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210247: Data Structures Lab

| **Teaching Scheme:** | **Credit** | **Examination Scheme:** |
|---|---|---|
| **PR: 04 Hours/Week** | **02** | **TW: 25 Marks** |
| | | **PR: 50 Marks** |

## Guidelines for Instructor's Manual

The instructor’s manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface etc), University syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

## Guidelines for Student Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, conclusion/analysis. **Program codes with sample output of all performed assignments are to be submitted as softcopy.**

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

## Guidelines for Assessment

Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

## Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student’s for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

## Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of Hungarian notation, proper indentation and comments. Use of open source software is to be encouraged.

In addition to these, instructor may assign one real life application in the form of a mini-project based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Set of suggested assignment list is provided in groups- A, B, C, D, and E. Each student must perform at least 13 assignments as at least 3 from group A, 3 from group B, 2 from group C, 2 from group D and 3 from group E.

**Operating System recommended** :- 64-bit Open source Linux or its derivative

**Programming tools recommended**: - Open Source C++ Programming tool like G++/GCC

## Suggested List of Laboratory Assignments

### Group A

1. In Second year Computer Engineering class of M students, set A of students play cricket and set B of students play badminton. Write C/C++ program to find and display-
   - i. Set of students who play either cricket or badminton or both
   - ii. Set of students who play both cricket and badminton
   - iii. Set of students who play only cricket
   - iv. Set of students who play only badminton
   - v. Number of students who play neither cricket nor badminton

   (Note- While realizing the set duplicate entries are to avoided)

2. Write C/C++ program to store marks scored for first test of subject 'Data Structures and Algorithms' for N students. Compute
   - I. The average score of class
   - ii. Highest score and lowest score of class
   - iii. Marks scored by most of the students
   - iv. list of students who were absent for the test

3. Department library has N books. Write C/C++ program to store the cost of books in array in ascending order. Books are to be arranged in descending order of their cost. Write function for
   - a) Reverse the contents of array without using temporary array.
   - b) Copy costs of books those with cost less than 500 in new array
   - c) Delete the duplicate entries using temporary array
   - d) Delete duplicate entries without using temporary array
   - e) Count number of books with cost more than 500.

4. Set A=(1,3, a, s, t, i} represent alphanumeric characters permitted to be used to set the password of length 4. Write C/C++ program to generate all possible passwords.

5. A magazine committee is to be formed that consists of any 3 members to be selected from { Nikhita, Aboli, Megha, Sanika, Pratik, Saurabh}. Write C/C++ program to list all possible committees.

6. It is decided that weekly greetings are to be furnished to wish the students having their birthdays in that week. The consolidated sorted list with desired categorical information is

to be provided to the authority. Write C++ program for array of structures to store students PRNs with date and month of birth. Let Array_A and Array_B be the two arrays for two SE Computer divisions. Arrays are sorted on date and month. Merge these two arrays into third array Array_SE_Comp_DOB resulting in sorted information about Date of Birth of SE Computer students.

7. A magic square is an $n * n$ matrix of the integers 1 to $n^2$ such that the sum of each row, column, and diagonal is the same. The figure given below is an example of magic square for case n=5. In this example, the common sum is 65. Write C/C++ Program for magic square.

| | | | | |
|---|---|---|---|---|
| **15** | **8** | **1** | **24** | **17** |
| **16** | **14** | **7** | **5** | **23** |
| **22** | **20** | **13** | **6** | **4** |
| **3** | **21** | **19** | **12** | **10** |
| **9** | **2** | **25** | **18** | **11** |

8. An m x n *matrix* is said to have a saddle point if some entry *a[i][j]* is the smallest value in row *i* and the largest value in *j*. Write C/ C++ function that determines the location of a saddle point if one exists.

9. Write C/C++ program for storing matrix. Write functions for
   a) Check whether given matrix is upper triangular or not
   b) Compute summation of diagonal elements
   c) Compute transpose of matrix
   d) Add, subtract and multiply two matrices

10. Write C++ program with class for String. Write a function

- *frequency* that determines the frequency of occurrence of particular character in the string.
- *delete* that accepts two integers, *start* and *length*. The function computes a new string that is equivalent to the original string, except that *length* characters being at *start* have been removed.
- *chardelete* that accepts a character *c*. The function returns the string with all occurrences of *c* removed.
- *replace* to make an in-place replacement of a substring *w* of a string by the string *x*. note that *w* may not be of same size of *x*
- *palindrome* to check whether given string is palindrome or not

11. Write C++ program for sparse matrix realization and operations on it- Transpose, Fast Transpose and addition of two matrices

12. Write C++ program for string operations- copy, concatenate, check substring, equal, reverse and length

13. Write a C++ program to realize polynomial equation and perform operations. Write function
   a) To input and output polynomials represented as $b_m x^{em} + b_{m-1} x^{em-1} + \ldots.. + b_0 x^{e0}$.
   Your functions should overload the << and >> operators.
   b) Evaluates a polynomial at given value of x
   c) Add two polynomials
   d) Multiplies two polynomials

## Group B

14. Department of Computer Engineering has student's club named 'Pinnacle Club'. Students of Second, third and final year of department can be granted membership on request. Similarly one may cancel the membership of club. First node is reserved for president of club and last node is reserved for secretary of club. Write C++ program to maintain club member's information using singly linked list. Store student PRN and Name. Write functions to

<table>
<tr><td></td><td>a) Add and delete the members as well as president or even secretary.<br>b) Compute total number of members of club<br>c) Display members<br>d) Display list in reverse order using recursion<br>e) Two linked lists exists for two divisions. Concatenate two lists.</td></tr>
<tr><td>15.</td><td>The ticket booking system of Cinemax theater has to be implemented using C++ program. There are 10 rows and 7 seats in each row. Doubly circular linked list has to be maintained to keep track of free seats at rows. Assume some random booking to start with. Use array to store pointers (Head pointer) to each row. On demand<br>a) The list of available seats is to be displayed<br>b) The seats are to be booked<br>c) The booking can be cancelled.</td></tr>
<tr><td>16.</td><td>Write C++ program for storing appointment schedule for day. Appointments are booked randomly using linked list. Set start and end time and min and max duration for visit slot. Write functions for-<br>a) Display free slots<br>b) Book appointment<br>c) Cancel appointment ( check validity, time bounds, availability etc)<br>d) Sort list based on time<br>e) Sort list based on time using pointer manipulation</td></tr>
<tr><td>17.</td><td>Second year Computer Engineering class, set A of students like Vanilla Ice-cream and set B of students like butterscotch ice-cream. Write C/C++ program to store two sets using linked list. compute and display-<br>i. Set of students who like either vanilla or butterscotch or both<br>ii. Set of students who like both vanilla and butterscotch<br>iii. Set of students who like only vanilla not butterscotch<br>iv. Set of students who like only butterscotch not vanilla<br>v. Number of students who like neither vanilla nor butterscotch</td></tr>
<tr><td>18.</td><td>Write C++ program to store set of negative and positive numbers using linked list. Write functions to<br>a) Insert numbers<br>b) Delete nodes with negative numbers<br>c) Create two more linked lists using this list, one containing all positive numbers and other containing negative numbers<br>d) For two lists that are sorted; Merge these two lists into third resultant list that is sorted</td></tr>
<tr><td>19.</td><td>Write C++ program for storing binary number using doubly linked lists. Write functions-<br>a) to compute 1’s and 2’s complement b) add two binary numbers</td></tr>
<tr><td>20.</td><td>Let x = ($x_1$,$x_2$, … , $x_n$) and y = ($y_1$, $y_2$,…. , $y_m$) be two doubly linked lists. Assume that in each linked list, the nodes are in non-decreasing order of their data-field values. Write C/C++ program to merge the two lists to obtain a new linked list z in which the nodes are also in this order. Following the merge, x and y should represent empty lists because each node initially in x or y is now in z. No additional nodes may be used.</td></tr>
<tr><td>21.</td><td>Design a linked allocation system to represent and manipulate univariate polynomials with integer coefficients (use circular linked lists with head nodes). Each term of the polynomial will be represented as a node Thus. a node in this system will have three data members as below:<br><table><tr><td>Exponent</td><td>Link</td></tr><tr><td colspan="2">Coefficient</td></tr></table></td></tr>
</table>

| | |
|---|---|
| | To erase polynomials efficiently, we need to use an available-space list and associated functions. The external (i.e.. for input or output) representation of a univariate polynomial will be assumed to be a sequence of integers of the form: $n, c_1, e_1 ,c_2. e_2. c_3. e_3 . . , c_n e_n$ where $e_i$ represents an exponent and $c_i$ a coefficient; n gives the number of terms in the polynomial. The exponents are in decreasing order — i.e., $e_1> e_2> …. >e_n$.<br>Write and test the following functions:<br>1. istream&**operator** >>(istream& is, Polynomial& x): Read in an input polynomial and convert it to its circular list representation using a head node.<br>2. ostream&**operator**<< (ostream&os, Polynomial& x): Convert x from its linked list representation to its external representation and output it.<br>3. Polynomial:: Polynomial(**const** Polynomial& a) [Copy Constructor]: Initialize the polynomial *this to the polynomial a.<br>4. **const** Polynomial& Polynomial :: **operator=**(**const** Polynomial& a) [Assignment Operator]: Assign polynomial a to*this.<br>5. Polynomial:: Polynomial ( ) [Destructor]: Return all nodes of the polynomial *this to the available-space list.<br>6. Polynomial **operator+** (**const** Polynomial& a, **const** Polynomial& b) [Addition]: Create and return the polynomial a + b. a and b are to be left unaltered.<br>7. Polynomial operator* (constPolynomial& a, constPolynomial& b) [Multiplication]: Create and return the polynomial a * b. a and b are to be left unaltered.<br>8. **floatPolynomial ::Evaluate(float x):** Evaluate the polynomial *this at x and return the result. |
| 22. | Write C++ program to realize Set using Generalized Liked List (GLL) Ex. A ={ a, b, {c, d,e, {}, {f,g}, h, I, {j,k}, l, m}. Store and print as set notation. |
| **Group C** | |
| 23. | A palindrome is a string of character that's the same forward and backward. Typically, punctuation, capitalization, and spaces are ignored. For example, "Poor Dan is in a droop" is a palindrome, as can be seen by examining the characters "poor danisina droop" and observing that they are the same forward and backward. One way to check for a palindrome is to reverse the characters in the string and then compare with them the original-in a palindrome, the sequence will be identical. Write C++ program with functions-<br>1. To check whether given string is palindrome or not that uses a stack to determine whether a string is a palindrome.<br>2. to remove spaces and punctuation in string, convert all the Characters to lowercase, and then call above Palindrome checking function to check for a palindrome<br>3. to print string in reverse order using stack |
| 24. | In any language program mostly syntax error occurs due to unbalancing delimiter such as (),{},[]. Write C++ program using stack to check whether given expression is well parenthesized or not. |
| 25. | Implement C++ program for expression conversion as infix to postfix and its evaluation using stack based on given conditions<br>i. Operands and operator, both must be single character.<br>ii. Input Postfix expression must be in a desired format.<br>iii. Only '+', '-', '*' and '/ ' operators are expected. |
| 26. | Implement C++ program for expression conversion-<br>a) infix to prefix, b)prefix to postfix,<br>c) prefix to infix, d) postfix to infix and e) postfix to prefix. |
| 27. | A classic problem that can be solved by backtracking is called the Eight Queens problem, which comes from the game of chess. The chess board consists of 64 square arranged in an |

| | |
|---|---|
| | 8 by 8 grid. The board normally alternates between black and white square, but this is not relevant for the present problem. The queen can move as far as she wants in any direction, as long as she follows a straight line, Vertically, horizontally, or diagonally. Write C++ program with recursive function for generating all possible configurations for 4-queen's problem. |
| **Group D** | |
| 28. | Queues are frequently used in computer programming, and a typical example is the creation of a job queue by an operating system. If the operating system does not use priorities, then the jobs are processed in the order they enter the system. Write C++ program for simulating job queue. Write functions to add job and delete job from queue. |
| 29. | Write program to implement a priority queue in C++ using an inorder List to store the items in the queue. Create a class that includes the data items(which should be template) and the priority (which should be int)The inorder list should contain these objects ,with operator <= overloaded so that the items with highest priority appear at the beginning of the list (which will make it relatively easy to retrieve the highest item.) |
| 30. | A double-ended queue(deque) is a linear list in which additions and deletions may be made at either end. Obtain a data representation mapping a deque into a one-dimensional array. Write C++ program to simulate deque with functions to add and delete elements from either end of the deque. |
| 31. | Pizza parlor accepting maximum M orders. Orders are served in first come first served basis. Order once placed cannot be cancelled. Write C++ program to simulate the system using circular queue using array. |
| **Group E** | |
| 32. | Write C++ program to store roll numbers of student in array who attended training program in random order. Write function for-<br>a) Searching whether particular student attended training program or not using linear search and sentinel search. b) Searching whether particular student attended training program or not using binary search and Fibonacci search. |
| 33. | Write C++ program to store names and mobile numbers of your friends in sorted order on names. a) Search your friend from list using binary search (recursive and non recursive). Insert friend if not present in phonebook. b) Search your friend from list using Fibonacci search. Insert friend if not present in phonebook. |
| 34. | Write C++ program to maintain club members, sort on roll numbers in ascending order. Write function ‘Ternary_Search’ to search whether particular student is member of club. Ternary search is modified binary search that divides array into 3 halves instead of two. |
| 35. | Write C++ program to store first year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using<br>a) Selection Sort b) Bubble sort and display top five scores. |
| 36. | Write C++ program to store second year percentage of students in array. Write function for sorting array of floating point numbers in ascending order using<br>a) Insertion sort b) Shell Sort and display top five scores. |
| 37. | Write C++ program to store first year percentage of students in array. Sort array of floating point numbers in ascending order using quick sort and display top five scores. |
| 38. | Write C++ program to store XII percentage of students in array. Sort array of floating point numbers in ascending order using bucket sort and display top five scores. |
| 39. | Write C++ program to store X percentage of students in array. Sort array of floating point numbers in ascending order using radix sort and display top five scores. |

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210248: Object Oriented Programming Lab

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| PR: 02 Hours/Week | 01 | TW: 25 Marks |
| | | PR: 50 Marks |

## Guidelines for Instructor's Manual

The instructor’s manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface etc), University syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

## Guidelines for Student Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- OOP feature/Concept in brief, algorithm, flowchart, test cases, conclusion/analysis. **Program codes with sample output of all performed assignments are to be submitted as softcopy.**

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

## Guidelines for Assessment

Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

## Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student’s for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

## Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. Encourage students for the use coding standards such as appropriate use of Hungarian notation, proper Indentation and comments. Use of open source software is encouraged. Instructor may also assign one real life application in the form of a mini-project. Based on the concepts learned.

Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.
**Operating System recommended** :- 64-bit Open source Linux or its derivative
**Programming tools recommended**: - Open Source C++ Programming tool like G++/GCC.
First assignment is compulsory. Set of suggested assignment list is provided in 3 groups- A, B, and C. Instructor is suggested to design assignments list by selecting/designing at least **12** suitable assignments from group A, B, and C- **compulsory assignment, 5** from group A, **4** from group B, **3** from group C.

## Suggested List of Laboratory Assignments

### Compulsory Assignment

**1.** Install, Configure 64 bit Linux Operating Systems, study basic architecture, memory system, and learn basic administration.

### Group A

2. Implement a class Complex which represents the Complex Number data type. Implement the following operations:
   1. Constructor (including a default constructor which creates the complex number 0+0i).
   2. Overloaded **operator+** to add two complex numbers.
   3. Overloaded **operator*** to multiply two complex numbers.
   4. Overloaded << and >> to print and read Complex Numbers.

**3.** Implement a class Quadratic that represents degree two polynomials i.e., polynomials of type $ax^2+bx+c$. The class will require three data members corresponding to a, b and c. Implement the following operations:
   1. A constructor (including a default constructor which creates the 0 polynomial).
   2. Overloaded **operator+** to add two polynomials of degree 2.
   3. Overloaded << and >> to print and read polynomials. To do this, you will need to decide what you want your input and output format to look like.
   4. A function eval that computes the value of a polynomial for a given value of x.
   5. A function that computes the two solutions of the equation $ax^2+bx+c=0$.

4. Implement a class CppArray which is identical to a one-dimensional C++ array (i.e., the index set is a set of consecutive integers starting at 0) except for the following :
   1. It performs range checking.
   2. It allows one to be assigned to another array through the use of the assignment operator (e.g. cp1= cp2)
   3. It supports a function that returns the size of the array.
   4. It allows the reading or printing of array through the use of **cout** and **cin**.

5. Write a C++ program create a calculator for an arithmetic operator (+, -, *, /). The program should take two operands from user and performs the operation on those two operands depending upon the operator entered by user. Use a switch statement to select the operation. Finally, display the result.
   Some sample interaction with the program might look like this:
   *Enter first number, operator, second number: 10 / 3*
   *Answer = 3.333333*
   *Do another (y/n)? y*
   *Enter first number, operator, second number: 12 + 100*
   *Answer = 112*
   *Do another (y/n)? n*

| | |
|---|---|
| 6. | Develop an object oriented program in C++ to create a database of student information system containing the following information: Name, Roll number, Class, division, Date of Birth, Blood group, Contact address, telephone number, driving license no. etc Construct the database with suitable member functions for initializing and destroying the data viz constructor, default constructor, Copy constructor, destructor, static member functions, friend class, this pointer, inline code and dynamic memory allocation operators-new and delete. |
| 7. | Create a class template to represent a generic vector. Include following member functions:<br>• To create the vector.<br>• To modify the value of a given element<br>• To multiply by a scalar value<br>• To display the vector in the form (10,20,30,…) |
| 8. | Create a class Rational Number (fractions) with the following capabilities:<br>a) Create a constructor that prevents a 0 denominator in a fraction, reduces or simplifies fractions that are not in reduced form and avoids negative denominators.<br>b) Overload the addition, subtraction, multiplication and division operators for this class.<br>c) Overload the relational and equality operators for this class. |
| 9. | Imagine a publishing company which does marketing for book and audiocassette versions. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float).<br>Write a program that instantiates the book and tape classes, allows user to enter data and displays the data members. If an exception is caught, replace all the data member values with zero values. |
| 10. | Write a function in C++ to count and display the number of lines not starting with alphabet 'A' present in a text file "STORY.TXT".<br>Example:<br>If the file "STORY.TXT" contains the following lines,<br>The roses are red.<br>A girl is playing there.<br>There is a playground.<br>An aeroplane is in the sky.<br>Numbers are not allowed in the password.<br>The function should display the output as 3. |
| 11. | Write C++ Program with base class convert declares two variables, val1 and val2, which hold the initial and converted values, respectively. It also defines the functions getinit( ) and getconv( ), which return the initial value and the converted value. These elements of convert are fixed and applicable to all derived classes that will inherit convert. However, the function that will actually perform the conversion, compute( ), is a pure virtual function that must be defined by the classes derived from convert. The specific nature of compute( ) will be determined by what type of conversion is taking place. |

| | |
|---|---|
| 12. | A book shop maintains the inventory of books that are being sold at the shop. The list includes details such as author, title, price, publisher and stock position. Whenever a customer wants a book, the sales person inputs the title and author and the system searches the list and displays whether it is available or not. If it is not, an appropriate message is displayed. If it is, then the system displays the book details and requests for the number of copies required. If the requested copies book details and requests for the number of copies required. If the requested copies are available, the total cost of the requested copies is displayed; otherwise the message “Required copies not in stock” is displayed. Design a system using a class called books with suitable member functions and Constructors. Use new operator in constructors to allocate memory space required. Implement C++ program for the system. |
| 13. | Create employee bio-data using following classes i) Personal record ii))Professional record iii) Academic record Assume appropriate data members and member function to accept required data & print bio-data. Create bio-data using multiple inheritance using C++. |
| **Group B** | |
| 14. | Crete User defined exception to check the following conditions and throw the exception if the criterion does not meet.<br>a. User has age between 18 and 55<br>b. User stays has income between Rs. 50,000 – Rs. 1,00,000 per month<br>c. User stays in Pune/ Mumbai/ Bangalore / Chennai<br>d. User has 4-wheeler<br>Accept age, Income, City, Vehicle from the user and check for the conditions mentioned above. If any of the condition not met then throw the exception. |
| 15. | Write a menu driven program that will create a data file containing the list of telephone numbers in the following form<br>John 23456<br>Ahmed 9876<br>……….. ………<br>Use a class object to store each set of data, access the file created and implement the following tasks<br>I. Determine the telephone number of specified person<br>II. Determine the name if telephone number is known<br>III. Update the telephone number, whenever there is a change. |
| 16. | Write a C++ program that creates an output file, writes information to it, closes the file and open it again as an input file and read the information from the file. |
| 17. | Write a C++ program using command line arguments to search for a word in a file and replace it with the specified word. The usage of the program is shown below.<br>*$ change <old word> <new word> <file name>* |
| 18. | Write a function template selection Sort. Write a program that inputs, sorts and outputs an integer array and a float array. |

19. You are the owner of a hardware store and need to keep an inventory that can tell you what different tools you have, how many of each you have on hand and the cost of each one. Write a program that initializes the random-access file hardware.dat to 100 empty records, lets you input the data concerning each tool, enables you to list all your tools, lets you delete a record for a tool that you no longer have and lets you update any information in the file. The tool identification number should be the record number. Use the following information to start your file:

| Record # | Tool name | Quantity | Cost |
|---|---|---|---|
| 3 | Electric sander | 7 | 57.98 |
| 17 | Hammer | 76 | 11.99 |
| 24 | Jig saw | 21 | 11.00 |
| 39 | Lawn mower | 3 | 79.50 |
| 56 | Power saw | 18 | 99.99 |

## Group C

20. Write C++ program using STL for implementation of Singly, doubly and circular linked list.

21. Write C++ program using STL for implementation of stack & queue using SLL

22. Write C++ program using STL to add binary numbers (assume one bit as one number); use STL stack.

23. Write C++ program using STL for Dqueue (Double ended queue)

24. Write C++ program using STL for Sorting and searching with user-defined records such as Person Record (Name, birth date, telephone no), item record (item code, item name, quantity and cost)

## Mini-projects

**25.** Design and develop the Tic-Tac-Toe Game using C++

26. Develop a Supermarket Billing System using C++. The key features of this application are listed below :

- **Bill Report**: It shows the bill report of all the items added in supermarket billing system.
- **Add, Remove or Edit items**: With this feature one can add, remove and modify item details. In add items, one can add information or details such as item no., item name, manufacturing date, price, quantity, tax percent, and many more.
- **Show item details**: This feature allows users to see the items and the corresponding details given for the item while adding the item.
  Use file to store the data.

27. Design an E-mail Verifier which accepts the email address from the user. Depending upon the input given by user display appropriate results. Use the following concepts in the Project – Constructor, Destructor, new, delete, exceptional handling, string handling functions, etc.

28. Design and Develop Library Management system using OOP Concepts.

29. Write a C++ program to implement a small database mini project to understand persistent objects and operations on sequential files (ex- library information, inventory systems, automated banking system, reservation systems etc.) For example, write a program to create a database for reservation system using information such as Name, sex, age, starting place of journey and destination. Program should have following facilities a) To display entire passenger list b) To display particular record c) To update record d) To delete and sort record. Use Exception Handling for data verification

## Savitribai Phule Pune University
## Second Year of Computer Engineering (2015 Course)
## 210249: Soft Skills

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| PR: 02 Hours /Week | 01 | TW: 25 Marks |

**Course Objectives:**

- To encourage the all round development of students by focusing on soft skills.
- To make the engineering students aware of the importance, the role and the content of soft skills through instruction, knowledge acquisition, demonstration and practice.
- To develop and nurture the soft skills of the students through individual and group activities.
- To expose students to right attitudinal and behavioral aspects and to build the same through activities

**Course Outcomes:**

On completion of the course, student will be able to–

- Effectively communicate through verbal/oral communication and improve the listening skills
- Write precise briefs or reports and technical documents.
- Actively participate in group discussion / meetings / interviews and prepare & deliver presentations.
- Become more effective individual through goal/target setting, self motivation and practicing creative thinking.
- Function effectively in multi-disciplinary and heterogeneous teams through the knowledge of team work, Inter-personal relationships, conflict management and leadership quality.

## Course Contents

### Unit I Self-Development

Introduction to soft skills, Self-Management: Self-Evaluation, Self-Discipline, Self-Criticism, Self-Awareness, Self-Esteem, Positive Thinking, Perceptions and Attitudes, Values and Belief Systems, Personal success factors, Handling failure, Knowing Yourself, identifying one's strengths and weaknesses, SWOT analysis, Johari's Window, Career Planning & Goal setting, prioritization, Managing self – emotions, ego, pride, stress; Personality development.

### Unit II Communication Skills

**Significance of Communication-** types, barriers of communication, effective communication, Verbal and non-verbal Communication, Speaking Skills – Importance of speaking effectively, speech process, message, audience, speech. Style, feedback, conversation and oral skills, fluency and self expression, body language phonetics and spoken English, speaking techniques, word stress, correct stress patterns, voice quality, correct tone, types of tones, positive image projection techniques, Public Speaking, Group discussion, Listening Skills: Virtues of Listening, Barriers and filters, Fundamentals of Good Listening, Reading Skills: Comprehension, reading research papers, Communication in a Digital World.

## Unit III Language and Writing Skills

Vocabulary: One - Word Substitutes, Words often Confused - Pairs of Words, Synonyms and Antonyms, Foreign Phrases, Phrasal verbs derived from the dynamic verbs, Business Writing: Note Making, Letter writing, Writing Formal Letters. Technical Report Writing, Memo, Notices/Circulars Agenda and Minutes of a Meeting, E-Mail, Essay writing. Employment Communication: Job Application, Preparation of CV and Resume writing. Presentation skills: Professional Presentation, Nature of Oral Presentation, Planning a Presentation, Preparing the Presentation, Delivering the Presentation.

## Unit IV Leadership and Team Building

Introduction, Leader and Leadership, Leadership Traits, Culture and Leadership: Salient Features of Corporate Culture, Leadership Styles, Leadership Trends, Team Building: Team Development Stages, Types of Teams: Cross-functional Team, Problem-solving Team, Inter- personal relations: Types of feelings, steps to deal with complex feelings. Assertiveness and Confidence building. Types of Conflict and resolutions. Emotions, emotional empathy and emotional intelligence.

## Unit V Stress and Time Management

Introduction, Stress in Today's Time: Identify the Stress Source, Signs of Stress, Ways to Cope with Stress : Healthier Ways to Combat Stress, Steps to be Taken in the Organizations : Open communication, Time Management, Working towards Your Goals, Smart Work, Prioritize your Tasks,4 Ds of Decision Making.

## Unit VI Ethics, Etiquette and Mannerism

Professional Etiquette: Etiquette at Meetings, Etiquette at Dining. Involuntary Awkward Actions, Public Relations Office(PRO)'s Etiquettes, Technology Etiquette : Phone Etiquette, Email Etiquette, Social Media Etiquette, Video Conferencing Etiquette, Interview Etiquette, Dressing Etiquettes : for Interview, offices and social functions, Ethical Values: Importance of Work Ethics, Problems in the Absence of Work Ethics.

## Books:

### Text:

1. Gajendra Singh Chauhan, Sangeeta Sharma: Soft Skills – An Integrated Approach to Maximize Personality, WILEY INDIA, ISBN:13:9788126556397.

### References:

1. Indrajit Bhattacharya, “An Approach to Communication Skills”, Delhi, Dhanpat Rai, 2008.
2. Simon Sweeney, “English for Business Communication”, Cambridge University Press, ISBN 13:978-0521754507.
3. Sanjay Kumar and Pushpa Lata, “Communication Skills”, Oxford University Press, ISBN 10:9780199457069.
4. Atkinson and Hilgard's, “Introduction to Psychology”, 14th Edition, Geoffrey Loftus, ISBN-10:0155050699 © 2003
5. Kenneth G. Mcgee, “Heads Up: How to Anticipate Business Surprises & Seize Opportunities First”, Harvard Business School Press, Boston, Massachusetts, 2004, ISBN 10:1591392993.
6. Krishnaswami, N. and Sriraman, T, “Creative English for Communication”, Macmillan.

## Guidelines for Instructor's Manual

The instructor’s manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/preface etc), University syllabus, conduction & Assessment guidelines, topics under consideration concept objectives, outcomes, guidelines, references.

## Guidelines for Student's Lab Journal and TW Assessment

The student must prepare the journal in the form of **report** elaborating the activities performed in the lab. Continuous assessment of laboratory work is to be done based on overall performance and lab assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, punctuality, neatness, enthusiasm, participation and contribution in various activities-SWOT analysis, presentations, team activity, event management, group discussion, Group exercises and interpersonal skills and similar other activities/assignments.

## Guidelines for Soft skills Lab Conduction

The instructor may frame assignments to enhance skills supporting career aspects. Multiple set of activity based assignments can be prepared and distributed among batches. Every student must be given adequate opportunity to participate actively in each activity. An exercise can be designed to allow multiple skills exposure for example a group task encouraging discussions, team building, value sharing, leadership and role play all at the same time.

## Suggested List of Laboratory Assignments

| | |
|---|---|
| **1.** | **SWOT analysis**<br>The students should be made aware of their goals, strengths and weaknesses, attitude, moral values, self confidence, etiquettes, non-verbal skills, achievements etc. through this activity. SWOT Analysis, Confidence improvement, values, positive attitude, positive thinking and self esteem. The concern teacher should prepare a questionnaire which evaluate students in all the above areas and make them aware about these aspects. |
| **2.** | **Personal & Career Goal setting – Short term & Long term**<br>The teacher should explain to them on how to set goals and provide template to write their short term and long term goals. |

| | |
|---|---|
| **3.** | **Public Speaking**<br>Any one of the following activities may be conducted :<br>**1. Prepared speech (**Topics are given in advance, students get 10 minutes to prepare the speech and 5 minutes to deliver.) **2. Extempore speech (**Students deliver speeches spontaneously for 5 minutes each on a given topic) **3. Story telling** (Each student narrates a fictional or real life story for 5 minutes each) **4. Oral review** ( Each student orally presents a review on a story or a book read by them) |
| **4.** | **Reading and Listening skills**<br>The batch can be divided into pairs. Each pair will be given an article (any topic) by the teacher. Each pair would come on the stage and read aloud the article one by one. After reading by each pair, the other students will be asked questions on the article by the readers. Students will get marks for correct answers and also for their reading skills. This will evaluate their reading and listening skills. The teacher should give them guidelines on improving their reading and listening skills. The teacher should also give passages on various topics to students for evaluating their reading comprehension. |
| **5.** | **Group discussion**<br>Group discussions could be done for groups of 5-8 students at a time Two rounds of a GD for each group should be conducted and teacher should give them feedback. |
| **6.** | **Letter/Application writing**<br>Each student will write one formal letter, and one application. The teacher should teach the students how to write the letter and application. The teacher should give proper format and layouts. |
| **7.** | **Report writing**<br>The teacher should teach the students how to write report .The teacher should give proper format and layouts. Each student will write one report based on visit / project / business proposal etc. |
| **8.** | **Resume writing-** Guide students and instruct them to write resume. |
| **9.** | **Presentation Skill**<br>Students should make a presentation on any informative topic of their choice. The topic may be technical or non-technical. The teacher should guide them on effective presentation skills. Each student should make a presentation for at least 10 minutes. |
| **10.** | **Team games for team building -** Students should make to participate in team activity. |
| **11.** | **Situational games for role playing as leaders** |
| **12.** | **Faculty may arrange one or more sessions from following:**<br>Yoga and meditation. Stress management, relaxation exercises, and fitness exercises.<br>Time management and personal planning sessions. |
| **13.** | **Mock interviews-** guide students and conduct mock interviews |

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210250: Audit Course 1

In addition to credits, it is recommended that there should be audit course in preferably in each semester from second year to supplement knowledge and skills. A student will be awarded the bachelor's degree if he/she earns 190 credits and clears all the audit courses specified in the syllabus. The student will be awarded grade as AP on successful completion of audit course.

The student may opt for one of the audit courses per semester, starting from second year first semester. Though not mandatory, such a selection of the audit courses helps the learner to explore the subject of interest in greater details resulting in achieving the very objective of audit course's inclusion. List of options offered is provided. Each student has to choose one audit course from the list per semester. Evaluation of audit course will be done at institute level itself. Method of conduction and method of assessment for audit courses are suggested.

## Criteria:

The student registered for audit course shall be awarded the grade AP (Audit Course Pass) and shall be included such AP grade in the Semester grade report for that course, provided student has the minimum attendance as prescribed by the Savitribai Phule Pune University and satisfactory in-semester performance and secured a passing grade in that audit course. No grade points are associated with this 'AP' grade and performance in these courses is not accounted in the calculation of the performance indices SGPA and CGPA. Evaluation of audit course will be done at institute level itself. (Ref- http://www.unipune.ac.in/Syllabi_PDF/revised-2015/engineering/UG_RULE_REGULATIONS_FOR_CREDIT_SYSTEM-2015_18June.pdf)

## Guidelines for Conduction and Assessment (Any one or more of following but not limited to)

- Lectures/ Guest Lectures
- Visits (Social/Field) and reports
- Demonstrations
- Surveys
- Mini Project
- Hands on experience on specific focused topic

## Guidelines for Assessment (Any one or more of following but not limited to)

- Written Test
- Demonstrations/ Practical Test
- Presentations
- IPR/Publication
- Report

## Audit Course 1 Options

| Course Code | Audit Course Title |
|---|---|
| **AC1-I** | Road Safety |
| **AC1-II** | Humanities and Social Sciences |
| **AC1-III** | Environmental Studies |
| **AC1-IV** | Smart Cities |
| **AC1-V** | Foreign Language (one of Japanese/Spanish/French/German). Course contents for **Japanese (Module 1)** are provided. For other languages institute may design suitably. |

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210250: Audit Course 1
# AC1-I: Road Safety

Road transport remains the least safe mode of transport, with road accidents representing the main cause of death of people. The boom in the vehicle population without adequate road infrastructure, poor attention to driver training and unsatisfactory regulation has been responsible for increase in the number of accidents. India's vehicle population is negligible as compared to the World statistics; but the comparable proportion for accidents is substantially large.

The need for stricter enforcement of law to ensure greater safety on roads and an environment-friendly road transport operation is of paramount importance. Safety and security are growing concerns for businesses, governments and the traveling public around the world, as also in India. It is, therefore, essential to take new initiatives in raising awareness, skill and knowledge of students as one of the ibid stake holders who are expected to follow the rules and policies of the government in order to facilitate safety of individual and safe mobility of others.

## Course Contents:

1. Existing Road Transport Scenario
2. Accident Causes & Remedies
3. Road Accident Investigation & Investigation Methods
4. Vehicle Technology – CMVR & Road Safety
5. Regulatory / Legislative Provisions for Improving Road Safety
6. Behavioral Training for Drivers for Improving Road Safety
7. Road Safety Education
8. Road Engineering Measures for Improving Road Safety

## References:

1. "Road Accidents in India Issues & Dimensions" , Ministry of Road Transport & Highways Government of India (www.unescap.org/sites/default/files/2.12.India_.pdf)
2. "Road Safety in India- Insights and analysis", http://indiatransportportal.com/wp-content/uploads/2012/11/Road_safety_2012.pdf
3. Road User's Handbook, ROADS & MARITIME PUBLICATIONS
4. "Improving Road Safety in Developing Countries", The national Academic Press

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210250: Audit Course 1
# AC1-II: Humanities and Social Sciences

Objective of Humanities and Social Science (HSS) is to produce well-rounded engineers, not only having good technological skills but also with the ability to interact with different organs of an organization.

HSS is concerned with society and the relationships among individuals within a society. It in turn has many branches, each of which is considered a "social science". The main social sciences include economics, political science, human geography, demography and sociology. In a wider sense, social science also includes some fields in the humanities such as anthropology, archaeology, psychology, history, law and linguistics.

## Course Objectives:

- Human and social development;
- Contemporary national and international affairs;
- Emergence of Indian society and Economics

## Course Outcomes:

On completion of the course, student will be able to–

- Making engineering and technology students aware of the various issues concerning man and society.
- These issues will help to sensitize students to be broader towards the social, cultural, economic and human issues, involved in social changes
- Able to understand the nature of the individual and the relationship between the self and the community
- Understanding major ideas, values, beliefs, and experiences that have shaped human history and cultures

## Course Contents

1. **Indian Society :** Structure of Indian Society, Indian Social Demography– Social and Cultural, Differentiations: caste, class, gender and tribe; Institutions of marriage, family and kinship- Secularization –Social Movements and Regionalism- Panchayatraj Institutions; Indian constitution; Affirmative Action Programme of the Government- various reservations and commissions.

2. **Social Development**: Scientific approach to the study of human beings. Evolution of human kind, social change and evolution. Industrial revolution. National policy on education, health and health care and human development.

3. **Sectoral Development: Agriculture:** Technology changes, Green revolutions, Employment Rural & Urban, Government Schemes. Industrial Development: Strategies, Public & Private Sectors, Categories, infrastructure, transport & communication, Consumer Awareness.

4. **Environment & Ecology:** Ecosystems: Structure, Working, components. Pollution: Water & Air Pollution, Global Warming, Control Strategies, International Treaties. Energy Sources: Renewable & Non Renewable, Hydro power, Biomass, Ocean, Geothermal & Tidal. Global Environmental Issues: Population Growth, Soil Degradation, Loss of Biodiversity.

## References:

1. Krugman, “ International Economics”, Pearson Education, ISBN-13:000-01334-23646
2. Prakash, “The Indian Economy”, Pearson Education, ISBN-8131758931
3. Thursen Gerald, “Engineering Economics”, Prentice Hall, ISBN-10:0138221227
4. C.S. Rao, “Environmental Pollution Control Engineering”, New Age International Pvt. Ltd, ISBN-812241835X
5. Rangarajan, “Environmental Issues in India, Pearson Education”, ISBN-10:8131708101
6. University of Delhi, “The Individual & Society”, Pearson Education. ISBN-8131704173
7. Wikipedia.org / wiki /social studies.
8. M. N. Srinivas, “Social change in modern India, 1991”, Orient Longman, ISBN-10:812500422X
9. David Mandelbaum, “Society in India”, 1990, Popular, ISBN-10:8171540139
10. David Newman, “Exploring the architecture of everyday life”, Pine Forge Press, 7th edition, ISBN-10:1452275947

# Savitribai Phule Pune University
## Second Year of Computer Engineering (2015 Course)
## 210250: Audit Course 1
## AC1-III: Environmental Studies

Environmental studies are the field that examines this relationship between people and the environment. An environmental study is an interdisciplinary subject examining the interplay between the social, legal, management, and scientific aspects of environmental issues.

### Course Objectives:

- Understanding the importance of ecological balance for sustainable development.
- Understanding the impacts of developmental activities and mitigation measures.
- Understand and realize the multi-disciplinary nature of the environment, its components, and inter-relationship between man and environment
- Understand the relevance and importance of the natural resources in the sustenance of life on earth and living standard

### Course Outcomes:

On completion of the course, student will be able to–

- Comprehend the importance of ecosystem and biodiversity
- To correlate the human population growth and its trend to the environmental degradation and develop the awareness about his/her role towards environmental protection and prevention
- Identify different types of environmental pollution and control measures
- To correlate the exploitation and utilization of conventional and non-conventional resources

### Course Contents:

1. **Natural Resources:** Introduction, Renewable and non-renewable, Forest, water, mineral, food, energy and land resources, Individual and conservation of resources, Equitable use of resources.
2. **Ecosystems:** Concept, Structure, Function, Energy flow, Ecological succession, Forest, grassland, desert and aquatic ecosystems - Introduction, characteristic features, structure and function.
3. **Biodiversity:** Genetic, Species and ecological diversity, Biogeographical classification of India, Value and hot spots, Biodiversity at global, national and local levels, India as mega-biodiversity nation, Threats to biodiversity, Endangered and endemic species of India, Conservation of Biodiversity, Endangered and endemic species, Conservation of biodiversity.
4. **Pollution:** Definition, Causes, effects and control measures of the pollution – Air, soil, Noise, Water, Marine and Thermal and Nuclear Pollution, Solid waste management, Role of Individual in Prevention of Pollution, Pollution case studies, Disaster management

### References:

1. Bharucha, E., “Textbook of Environmental Studies”, Universities Press (2005), ISBN-10:8173715408
2. Mahua Basu, “Environmental Studies”, Cambridge University Press, ISBN-978-1-107-5317-3

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210250: Audit Course 1
# AC1-IV: Smart Cities

We breathe in a world defined by urbanization and digital ubiquity, where mobile broadband connections outnumber fixed ones, machines dominate a new "internet of things," and more people live in cities than in the countryside. This course enables us to take a broad historical look at the forces that have shaped the planning and design of cities and information technologies from the rise of the great industrial cities of the nineteenth century to the present. This course considers the motivations, aspirations, and shortcomings of them all while offering a new civics to guide our efforts as we build the future together, one click at a time.

## Course Objectives:

- To identify urban problems
- To study Effective and feasible ways to coordinate urban technologies.
- To study models and methods for effective implementation of Smart Cities.
- To study new technologies for Communication and Dissemination.
- To study new forms of Urban Governance and Organization.

## Course Outcomes:

On completion of the course, learner will be able to–

- Better understanding of the dynamic behavior of the urban system by going beyond the physical appearance and by focusing on representations, properties and impact factors
- Exploration of the city as the most complex human-made organism with a metabolism that can be modeled in terms of stocks and flows
- Knowledge about data-informed approaches for the development of the future city, based on crowd sourcing and sensing
- Knowledge about the latest research results in for the development and management of future cities
- Understanding how citizens can benefit from data-informed design to develop smart and responsive cities

## Course Contents:

Urbanization and Ubiquity - The slow emergence of learning cities in an urbanizing world Cities as collective learners, what do we know?- Framing a view -A gamut of learning types - Secrets of knowing and accelerating change - Why some cities learn and others do not.

## References:

1. Anthony M. Townsend, W. W. Norton & Company “Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia”, ISBN: 0393082873, 9780393082876.
2. Tim Campbell, Routledge “Beyond Smart Cities: How Cities Network, Learn and Innovate”, Routledge, ISBN: 9781849714266.
3. Stan Geertman, Joseph Ferreira, Jr. Robert Goodspeed, John Stillwell, “Planning Support System ms and Smart Cities”, Lecture notes in Geo information and Cartography, Springer.

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210250: Audit Course 1
# AC1-V: Foreign Language- Japanese (Module 1)

## About course:

With changing times, the competitiveness has gotten into the nerves and 'Being the Best' at all times is only the proof of it. Nonetheless, 'being the best' differs significantly from 'Communicating the best'. The best can merely be communicated whilst using the best suited Language!

Japanese is the new trend of 21st century. Not only youngsters but even the professionals seek value in it. It is the engineer's companion in current times with an assertion of a thriving future. Pune has indisputably grown to become a major center of Japanese Education in India while increasing the precedence for Japanese connoisseurs.

Japanese certainly serves a great platform to unlock a notoriously tough market & find a booming career. While the companies prefer candidates having the knowledge of the language, it can additionally help connect better with the native people thus prospering in their professional journey. Learning Japanese gives an extra edge to the 'resume' since the recruiters consciously make note of the fact it requires real perseverance and self-discipline to tackle one of the most complex languages.

It would be easy for all time to quit the impossible; however it takes immense courage to reiterate the desired outcomes, recognize that improvement is an ongoing process and ultimately soldier on it. The need of an hour is to introduce Japanese language with utmost professionalism to create awareness about the bright prospects and to enhance the proficiency and commitment. It will then prove to be the ultimate path to the quest for professional excellence!

## Course Objectives:

- To meet the needs of ever growing industry with respect to language support.
- To get introduced to Japanese society and culture through language.

## Course Outcomes:

On completion of the course student

- will have ability of basic communication.
- will have the knowledge of Japanese script.
- will get introduced to reading , writing and listening skills
- will develop interest to pursue professional Japanese Language course.

## Course Contents:

1 : Introduction to Japanese Language. Hiragana basic Script, colors, Days of the week

2 : Hiragana : modified Kana, double consonant, Letters combined with ya, yu, yo
Long vowels, Greetings and expressions

3 : Self Introduction, Introducing other person, Numbers, Months, Dates, Telephone numbers, Stating one's age.

## References:

1. Minna No Nihongo, "Japanese for Everyone", Elementary Main Text book 1-1 (Indian Edition), Goyal Publishers & Distributors Pvt. Ltd.
2. http://www.tcs.com (http://www.tcs.com/news_events/press_releases/Pages/TCS-Inaugurates-Japan-centric-Delivery-Center-Pune.aspx)

# SEMESTER II

# Savitribai Phule Pune University
## Second Year of Computer Engineering/IT (2015 Course)
## 207003: Engineering Mathematics III

| Teaching Scheme: | | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | Credit | In-Sem(online): 50 Marks |
| TUT: 01 Hour/Week | 05 | End-Sem(paper): 50 Marks |
| | | TW: 25 Marks |

### Prerequisites:

Differential and Integral Calculus, Taylor series and Infinite series, Differential equations of first order and first degree, Fourier series, Measures of Central tendency and dispersion, Vector algebra, Algebra of complex numbers.

### Course Objectives:

After completing this course, student will have adequate mathematical background, conceptual clarity, computational skills and algorithm design for problem solving related to:

- Linear differential equations of higher order applicable to Control systems, Computer vision and Robotics.
- Transform techniques such as Fourier transform, Z-transform and applications to Image processing.
- Statistical methods such as correlation, regression analysis and probability theory to analyze data and to make predictions applicable to machine intelligence.
- Vector calculus necessary to analyze and design complex electrical and electronic devices as appropriate to Computer engineering.
- Complex functions, conformal mappings and contour integration applicable to Image processing, Digital filters and Computer graphics.

### Course Outcomes:

On completion of the course, student will be able to–

- Solve higher order linear differential equation using appropriate techniques for modeling and analyzing electrical circuits.
- Solve problems related to Fourier transform, Z-Transform and applications to Signal and Image processing.
- Apply statistical methods like correlation, regression analysis and probability theory for analysis and prediction of a given data as applied to machine intelligence.
- Perform vector differentiation and integration to analyze the vector fields and apply to compute line, surface and volume integrals.
- Analyze conformal mappings, transformations and perform contour integration of complex functions required in Image processing, Digital filters and Computer graphics.

## Course Contents

| Unit I | Linear Differential Equations (LDE) and Applications | 09 Hours |
|---|---|---|

LDE of $n^{th}$ order with constant coefficients, Method of variation of parameters, Cauchy’s & Legendre’s DE, Simultaneous & Symmetric simultaneous DE. Modeling of Electrical circuits.

| Unit II | Transforms | 09 Hours |
|---|---|---|

**Fourier Transform (FT):** Complex exponential form of Fourier series, Fourier integral theorem, Fourier Sine & Cosine integrals, Fourier transform, Fourier Sine and Cosine transforms and their inverses, Discrete Fourier Transform.

**Z - Transform (ZT):** Introduction, Definition, Standard properties, ZT of standard sequences and their inverses. Solution of difference equations.

| Unit III | Statistics | 09 Hours |
|---|---|---|

Measures of central tendency, Standard deviation, Coefficient of variation, Moments, Skewness and Kurtosis, Curve fitting: fitting of straight line, parabola and related curves, Correlation and Regression, Reliability of Regression Estimates.

| Unit IV | Probability and Probability Distributions | 09 Hours |
|---|---|---|

Probability, Theorems on Probability, Bayes Theorem, Random variables, Mathematical Expectation, Probability density function, Probability distributions: Binomial, Poisson, Normal and Hypergometric, Test of Hypothesis: Chi-Square test, t-distribution.

| Unit V | Vector Calculus | 09 Hours |
|---|---|---|

Vector differentiation, Gradient, Divergence and Curl, Directional derivative, Solenoid and Irrigational fields, Vector identities. Line, Surface and Volume integrals, Green’s Lemma, Gauss’s Divergence theorem and Stoke’s theorem.

| Unit VI | Complex Variables | 09 Hours |
|---|---|---|

Functions of Complex variables, Analytic functions, Cauchy-Riemann equations, Conformal mapping, Bilinear transformation, Cauchy’s integral theorem, Cauchy’s integral formula, Laurent’s series, and Residue theorem.

## Books:

### Text:

1. Erwin Kreyszig, “Advanced Engineering Mathematics”. 9e, Wiley India, ISBN - 10: 9780470458365, 13: 978-0470458365.
2. Peter V. O'Neil, “Advanced Engineering Mathematics”, 7e, Cengage Learning, ISBN - 10: 1111427410.

### References:

1. M. D. Greenberg, “Advanced Engineering Mathematics”, 2e, Pearson Education, ISBN: 10: 0133214311, 13: 978-0133214314.
2. Wylie C.R. & Barrett L.C., “Advanced Engineering Mathematics”, McGraw-Hill, Inc., ISBN: 0-07-072188-2.
3. B. S. Grewal, “ Higher Engineering Mathematics”, Khanna Publication, Delhi, ISBN : 92 803-11026
4. P. N. Wartikar & J. N. Wartikar,” Applied Mathematics”, Vidyarthi Griha Prakashan, Pune, ISBN: 0-7923-0594-9.
5. B.V. Ramana, “Higher Engineering Mathematics”, Tata McGraw-Hill, ISBN: 007063419X.
6. Thomas L. Harman, Dabney and Norman Richert, “Advanced Engineering Mathematics with MATLAB”, 2e, Thomson Learning, ISBN: 13: 978-0534371647

## Guidelines for Tutorial and Term Work:

- Tutorial shall be engaged in four batches (batch size of 20 students maximum) per division
- Term work shall be based on continuous assessment of six assignments (one per each unit) and performance in internal tests

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210251: Computer Graphics

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | 04 | In-Sem(online): 50 Marks<br>End-Sem(paper): 50 Marks |

**Prerequisite:**

- Data Structures and algorithms
- Basic Mathematics, Geometry, linear algebra, vectors and matrices.

**Course Objectives:**

- To acquaint the learner with the basic concepts of Computer Graphics
- To learn the various algorithms for generating and rendering graphical figures
- To get familiar with mathematics behind the graphical transformations
- To understand and apply various methods and techniques regarding projections, animation, shading, illumination and lighting

**Course Outcomes:**

On completion of the course, student will be able to–

- Apply mathematics and logic to develop Computer programs for elementary graphic operations
- Develop scientific and strategic approach to solve complex problems in the domain of Computer Graphics
- Develop the competency to understand the concepts related to Computer Vision and Virtual reality
- Apply the logic to develop animation and gaming programs

## Course Contents

| Unit I | Graphics Primitives and Scan Conversion | 09 Hours |
|---|---|---|

Concepts, applications of computer graphics, pixel, frame buffer, resolution, aspect ratio.

**Plotting Primitives**: Scan conversions, lines, line segments, vectors, pixels and frame buffers, vector generation

**Scan Conversion:** Line and line segments, qualities of good line drawing algorithms,
line drawing algorithms: Digital Differential Analyzer (DDA), Bresenham and parallel line algorithms, Line styles: thick, dotted and dashed. Circle drawing algorithm: DDA, Bresenham. Character generating methods: stroke and bitmap method.

**Display Files**: display file structure, algorithms and display file interpreter. Primitive operations on display file.

| Unit II | Polygons and Clipping Algorithms | 09 Hours |
|---|---|---|

Introduction to polygon, types: convex, concave and complex. Representation of polygon, Inside test, polygon filling algorithms – flood fill, seed fill, scan line fill and filling with patterns.

**Windowing and clipping:** viewing transformations, 2-D clipping: Cohen – Sutherland algorithm, Polygon clipping: Sutherland Hodgeman algorithm, generalized clipping.

## Unit III — 2-D, 3-D Transformations and Projections — 09 Hours

**2-D transformations:** introduction, matrices, Translation, scaling, rotation, homogeneous coordinates and matrix representation, translation, coordinate transformation, rotation about an arbitrary point, inverse and shear transformation.

**3-D transformations:** introduction, 3-D geometry, primitives, 3-D transformations and matrix representation, rotation about an arbitrary axis, 3-D viewing transformations, 3-D Clipping

**Projections :** Parallel (Oblique: Cavalier, Cabinet and orthographic: isometric, diametric, trimetric) and Perspective ( Vanishing Points – 1 point, 2 point and 3 point)

## Unit IV — Segment and Animation — 09 Hours

**Segment:** Introduction, Segment table, Segment creation, closing, deleting and renaming, Visibility. **Animation:** Introduction, Design of animation sequences, Animation languages, Key-frame, Morphing, Motion specification.

**Colour models and applications:** Properties of Light, CIE chromaticity Diagram, RGB, HSV, CMY, YIQ, colour Selection and applications.

## Unit V — Shading, and Hidden Surfaces — 09 Hours

**Illumination Models:** Light Sources, Ambient Light, Diffuse reflection, Specular Reflection, and the Phong model, Combined diffuse and Specular reflections with multiple light sources, warn model, **Shading Algorithms:** Halftone, Gauraud and Phong Shading. **Hidden Surfaces** Introduction, Back face detection and removal, Algorithms: Depth buffer (z), Depth sorts (Painter), Area subdivision (Warnock), BSP tree, and Scan line.

## Unit VI — Curves and Fractals — 09 Hours

**Curves**: Introduction, Interpolation and Approximation, Blending function, B-Spline curve, Bezier curve, **Fractals**: Introduction, Classification, Applications, Fractal generation: snowflake, Triadic curve, Hilbert curve. **Gaming:** Introduction, Gaming platform (NVIDIA, i8060 etc.), Advances in Gaming, **Graphics Tools:** Introduction, Interactive graphics tool: OpenGL

## Books:

### Text:

1. S. Harrington, "Computer Graphics", 2nd Edition, McGraw-Hill Publications, 1987, ISBN 0 – 07 – 100472 – 6.
2. D. Rogers, "Procedural Elements for Computer Graphics", 2nd Edition, Tata McGraw-Hill Publication, 2001, ISBN 0 – 07 – 047371 – 4.
3. Donald D. Hearn, "Computer Graphics with Open GL", 4th Edition, ISBN-13: 9780136053583.

### References:

1. J. Foley, V. Dam, S. Feiner, J. Hughes, "Computer Graphics Principles and Practice", 2nd Edition, Pearson Education, 2003, ISBN 81 – 7808 – 038 – 9.
2. D. Rogers, J. Adams, "Mathematical Elements for Computer Graphics", 2nd Edition, Tata McGrawHill Publication, 2002, ISBN 0 – 07 – 048677 – 8.
3. Mario Zechner, Robert Green, "Beginning Android 4 Games Development", Apress, ISBN: 978-81- 322-0575-3.

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210252: Advanced Data Structures

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | 04 | In-Sem(online): 50 Marks<br>End-Sem(paper): 50 Marks |

## Prerequisite:

- Data Structures and algorithms
- Basic Mathematics, Geometry, linear algebra, vectors and matrices.

## Course Objectives:

- To develop a logic for graphical modelling of the real life problems.
- To suggest appropriate data structure and algorithm for graphical solutions of the problems.
- To understand advanced data structures to solve complex problems in various domains.
- To operate on the various structured data
- To build the logic to use appropriate data structure in logical and computational solutions.
- To understand various algorithmic strategies to approach the problem solution.

## Course Outcomes:

On completion of the course, student will be able to–

- To apply appropriate advanced data structure and efficient algorithms to approach the problems of various domain.
- To design the algorithms to solve the programming problems.
- To use effective and efficient data structures in solving various Computer Engineering domain problems.
- To analyze the algorithmic solutions for resource requirements and optimization
- To use appropriate modern tools to understand and analyze the functionalities confined to the data structure usage.

## Course Contents

| Unit I | Trees | 09 Hours |
|---|---|---|

**Tree-** basic terminology, General tree and its representation, representation using sequential and linked organization, Binary tree- properties, converting tree to binary tree, **binary tree traversals-** inorder, preorder, post order, level wise -depth first and breadth first, Operations on binary tree. Binary Search Tree (BST), BST operations, Threaded binary tree- concepts, threading, insertion and deletion of nodes in in-order threaded binary tree, in order traversal of in-order threaded binary tree. **Case Study**- Use of binary tree in expression tree-evaluation and Huffman's coding

| Unit II | Graphs | 09 Hours |
|---|---|---|

Basic Concepts, Storage representation, Adjacency matrix, adjacency list, adjacency multi list, inverse adjacency list. Traversals-depth first and breadth first, Introduction to Greedy Strategy, Minimum spanning Tree, Greedy algorithms for computing minimum spanning tree- Prims and Kruskal Algorithms, Dikjtra's Single source shortest path, Topological ordering.

**Case study-** Data structure used in Webgraph and Google map.

### Unit III — Hashing — 09 Hours

**Hash Table-** Concepts-hash table, hash function, bucket, collision, probe, synonym, overflow, open hashing, closed hashing, perfect hash function, load density, full table, load factor, rehashing, issues in hashing, hash functions- properties of good hash function, division, multiplication, extraction, mid-square, folding and universal, Collision resolution strategies- open addressing and chaining, Hash table overflow- open addressing and chaining, extendible hashing.

**Dictionary-** Dictionary as ADT, ordered dictionaries.

**Skip List-** representation, searching and operations- insertion, removal.

### Unit IV — Search Trees — 09 Hours

**Symbol Table-**Representation of Symbol Tables- Static tree table and Dynamic tree table, Introduction to Dynamic Programming, Weight balanced tree, Optimal Binary Search Tree (OBST), OBST as an example of Dynamic Programming, Height Balanced Tree- AVL tree.

### Unit V — Indexing and Multiway Trees — 09 Hours

**Indexing and Multiway Trees-** Indexing, indexing techniques, Types of search tree- Multiway search tree, B-Tree, B+Tree, Trie Tree, Splay Tree, Red-Black Tree, K-dimensional tree, AA tree.

**Set-** Set ADT, realization of Set and operations.

**Heap-**Basic concepts, realization of heap and operations, Heap as a priority queue, heap sort

### Unit VI — File Organization — 09 Hours

**Sequential file organization-** concept and primitive operations, **Direct Access File-** Concepts and Primitive operations, **Indexed sequential file organization-**concept, types of indices, structure of index sequential file, **Linked Organization-** multi list files, coral rings, inverted files and cellular partitions.

**External Sort-** Consequential processing and merging two lists, multiday merging- a k way merge algorithm.

### Books:

**Text:**

1. Horowitz, Sahani, Dinesh Mehata, “Fundamentals of Data Structures in C++”, Galgotia Publisher, ISBN: 8175152788, 9788175152786.
2. M Folk, B Zoellick, G. Riccardi, “File Structures”, Pearson Education, ISBN:81-7758-37-5
3. Peter Brass, “Advanced Data Structures”, Cambridge University Press, ISBN: 978-1-107-43982-5

**References:**

1. A. Aho, J. Hopcroft, J. Ulman, “Data Structures and Algorithms”, Pearson Education, 1998, ISBN-0-201-43578-0.
2. Michael J Folk, “File Structures an Object Oriented Approach with C++”, Pearson Education, ISBN: 81-7758-373-5.
3. Sartaj Sahani, “Data Structures, Algorithms and Applications in C++”, Second Edition, University Press, ISBN:81-7371522 X.
4. G A V Pai, “Data Structures and Algorithms”, The McGraw-Hill Companies, ISBN - 9780070667266.
5. Goodrich, Tamassia, Goldwasser, “Data Structures and Algorithms in Java”, Wiley Publication, ISBN: 9788126551903.

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210253: Microprocessor

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 04 Hours/Week | 04 | In-Sem(online): 50 Marks<br>End-Sem(paper): 50 Marks |

**Prerequisite:** Digital Electronics and Logic Design

## Course Objectives:

- To learn the architecture and programmer's model of advanced processor
- To understand the system level features and processes of advanced processor
- To acquaint the learner with application instruction set and logic to build assembly language programs.
- To understand debugging and testing techniques confined to 80386 DX

## Course Outcomes:

On completion of the course, student will be able to–

- To apply the assembly language programming to develop small real life embedded application.
- To understand the architecture of the advanced processor thoroughly to use the resources for programming
- To understand the higher processor architectures descended from 80386 architecture

## Course Contents

| Unit I | 80386DX- Basic Programming Model and Applications Instruction Set | 09 Hours |
|---|---|---|

**Memory Organization and Segmentation-** Global Descriptor Table, Local Descriptor Table, Interrupt Descriptor Table, Data Types, Registers, Instruction Format, Operand Selection, Interrupts and Exceptions

**Applications Instruction Set-** Data Movement Instructions, Binary Arithmetic Instructions, Decimal Arithmetic Instructions, Logical Instructions, Control Transfer Instructions, String and Character Transfer Instructions, Instructions for Block Structured Language, Flag Control Instructions, Coprocessor Interface Instructions, Segment Register Instructions, Miscellaneous Instructions.

| Unit II | Systems Architecture and Memory Management | 09 Hours |
|---|---|---|

**Systems Architecture-** Systems Registers, Systems Instructions.

**Memory Management-** Segment Translation, Page Translation, Combining Segment and Page Translation.

| Unit III | Protection and Multitasking | 09 Hours |
|---|---|---|

**Protection-** Need of Protection, Overview of 80386DX Protection Mechanisms, Segment Level Protection, Page Level Protection, Combining Segment and Page Level Protection.

**Multitasking-** Task State Segment, TSS Descriptor, Task Register, Task Gate Descriptor, Task Switching, Task Linking, Task Address Space.

### Unit IV Input-Output, Exceptions and Interrupts 09 Hours

**Input-Output-** I/O Addressing, I/O Instructions, Protection and I/O

**Exceptions and Interrupts-** Identifying Interrupts, Enabling and Disabling Interrupts, Priority among Simultaneous Interrupts and Exceptions, Interrupt Descriptor Table (IDT), IDT Descriptors, Interrupt Tasks and Interrupt Procedures, Error Code, and Exception Conditions.

### Unit V Initialization of 80386DX, Debugging and Virtual 8086 Mode 09 Hours

**Initialization-** Processor State after Reset, Software Initialization for Real Address Mode, Switching to Protected Mode, Software Initialization for Protected Mode, Initialization Example, TLB Testing

**Debugging-** Debugging Features of the Architecture, Debug Registers, Debug Exceptions, Breakpoint Exception

**Virtual 8086 Mode-** Executing 8086 Code, Structure of V86 Stack, Entering and Leaving Virtual 8086 Mode.

### Unit VI 80386DX Signals, Bus Cycles and 80387 Coprocessor 09 Hours

**80386DX Signals-** Signal Diagram, Description of Signals **80386DX Bus Cycles-** System Clock, Bus States, Pipelined and Non-pipelined Bus Cycles.

**80387 NDP-** Control Register bits for Coprocessor support, 80387 Register Stack, Data Types, Load and Store Instructions, Trigonometric and Transcendental Instructions, Interfacing signals of 80386DX with 80387.

### Books:

**Text:**

1. Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630-011, December 1995.
2. James Turley, “Advanced 80386 Programming Techniques”, McGraw-Hill, ISBN: 10: 0078813425, 13: 978-0078813429.
3. Intel 387DX Math coprocessor, Order no.: 240448-005, March1992.

**References:**

1. Chris H. Pappas, William H. Murray, “80386 Microprocessor Handbooks”, McGraw-Hill Osborne Media, ISBN-10: 0078812429, 13: 978-0078812422.
2. Walter A. Triebel, “The 80386Dx Microprocessor: Hardware”, Software, and Interfacing, Pearson Education, ISBN: 0137877307, 9780137877300.
3. Brey, Barry B, “8086/8088, 80286, 80386 and 80486 Assembly Language Programming”, Prentice Hall, ISBN: 13: 9780023142475.
4. Mohammad Rafiquzzaman, “Microprocessors: Theory and Applications: Intel and Motorola", Prentice Hall, ISBN:-10:0966498011, 13:978:0966498011.
5. K. Bhurchandi, A. Ray, “Advanced Microprocessors and Peripherals”, McGraw Hill Education, Third Edition, ISBN: 978-1-25-900613-5.
6. Introduction to 64 bit Intel Assembly Language Programming for Linux, 2nd Edition, Ray Seyfarth, ISBN10: 1478119209, ISBN-13: 9781478119203, 2012.
7. Assembly Language Step-by-step: Programming with Linux, 3rd Edition, Jeff Duntemann, Wiley ISBN:-10 0470497025, ISBN-13: 978-0470497029, 2009.

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210254: Principles of Programming Languages

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| TH: 03 Hours/Week | TH: 03 | In-Sem(online): 50 Marks<br>End-Sem(paper): 50 Marks |

**Prerequisite:** Data Structures & Algorithms and Object Oriented Programming

## Course Objectives:

- To learn principles of programming language
- To understand structural, computational and logical implications regarding programming languages
- To explore main programming paradigms
- To understand and apply Object Oriented Programming (OOP) principles using C++ and Java

## Course Outcomes:

On completion of the course, student will be able to–

- To analyze the strengths and weaknesses of programming languages for effective and efficient program development.
- To inculcate the principles underlying the programming languages enabling to learn new programming languages.
- To grasp different programming paradigms
- To use the programming paradigms effectively in application development.

## Course Contents

| Unit I | Programming Language Syntax and semantics | 07 Hours |
|---|---|---|

Software development process, language and software development environments, language and software design methods, languages and computer architecture, programming language qualities, languages and reliability, languages and maintainability, languages and efficiency, a brief historical perspective and early high level languages, a bird's eye view of programming language concepts. **Syntax and semantics**-language definition, syntax, abstract syntax, concrete syntax, and pragmatics, semantics, an introduction to formal semantics, languages, language processing, interpretation, translation, the concept of binding, variables, name and scope, Type, l-value, r-value, reference and unnamed variables, routines, generic routines, aliasing and overloading, an abstract semantic processor, run time structure. Case study- run time structure of C.

| Unit II | Structuring the Data, Computations and Program | 07 Hours |
|---|---|---|

**Structuring of Data-** Built in and primitive types, Data aggregates and type constructors, Cartesian product, Finite mapping User-defined types and abstract data types, Type systems, Static versus dynamic program checking, Strong typing and type checking, Type compatibility, Type conversions, Types and subtypes, Generic types, monomorphic versus polymorphic type systems, Case Study- The type structure of C++, Java. **Structuring of Computations-** Structuring the computation, Expressions and statements, Conditional execution and iteration, Routines, Style issues: side effects and aliasing, Exceptions, Case Study-Exception handling in C++.

| Unit III | Structuring of Program | 07 Hours |
|---|---|---|

**Structuring of Program-**
Software design method, Concepts in support of modularity, Encapsulation, Interface and implementation, Separate and independent compilation, Libraries of modules, Language features for programming in the large, Program organization, Grouping of units, Encapsulation, Interface and implementation, Abstract data types, classes, and modules, Generic units, Generic data structures, Generic algorithms, Generic modules, Higher levels of genericity.

**Programming paradigms-** Introduction to programming paradigms, Introduction to four main Programming paradigms- procedural, object oriented, functional, and logic & rule based.

**Study of Java as Object oriented programming language.**

| Unit IV | Java as Object Oriented Programming Language-Overview | 07 Hours |
|---|---|---|

Java History, Java Features, Java and Internet, Java and Word Wide Web, Web Browsers, Java Virtual Machine, **Data Types and Size** (Signed vs. Unsigned, User Defined vs. Primitive Data Types, Explicit Pointer type) **Arrays:** one dimensional array, multi-dimensional array, alternative array declaration statements. **Control Statements** Revision of identical selection Statements in brief (if, else if, Nested if, Switch, Nested Switch), Iterative Statements For Each version of For Loop, Declaring Loop Control Variables Inside the for loop, Using comma in for loop), Jump Statements (Labeled Break and Labeled Continue), **String Handling:** String class methods.

| Unit V | Inheritance, Polymorphism, Encapsulation using Java | 07 Hours |
|---|---|---|

**Classes and Methods:** class fundamentals, declaring objects, assigning object reference variables, adding methods to a class, returning a value, constructors, this keyword, garbage collection, finalize() method, overloading methods, argument passing, object as parameter, returning objects, access control, static, final, nested and inner classes, command line arguments, variable-length arguments. **Inheritances**: member access and inheritance, super class references, Using super, multilevel hierarchy, constructor call sequence, method overriding, dynamic method dispatch, abstract classes, Object class. **Packages and Interfaces:** defining a package, finding packages and CLASSPATH, access protection, importing packages, interfaces (defining, implementation, nesting, applying), variables in interfaces, extending interfaces, instance of operator.

| Unit VI | Exception Handling in Java | 07 Hours |
|---|---|---|

fundamental, exception types, uncaught exceptions, try, catch, throw, throws, finally, multiple catch clauses, nested try statements, built-in exceptions, custom exceptions (creating your own exception sub classes). **Managing I/O:** Streams, Byte Streams and Character Streams, Predefined Streams, Reading console Input, Writing Console Output, Print Writer class, **Applet**: Applet Fundamental, Applet Architecture, Applet Skeleton, Requesting Repainting, status window, HTML Applet tag, passing parameters to Applets, Difference between Applet and Application Program.

## Books:

### Text:

1. Carlo Ghezzi, Mehdi Jazayeri, “Programming Language Concepts”,3rd Ed, Wiley Publication ISBN : 978-81-265-1861-6.
2. Herbert Schildt, "The Complete Reference Java", 9th Ed, TMH,ISBN: 978-0-07-180856-9.

### References:

1. Sebesta R., "Concepts of Programming Languages", 4th Edition, Pearson Education, ISBN-81-7808-161-X.
2. Deugo, “Java Gems”, Cambridge University Press, ISBN 10: 0521648246 ISBN 13: 9780521648240
3. T. W. Pratt, M. V. Zelkowitz, "Programming Languages Design and Implementation”, 4th Ed, PHI, ISBN 81-203-2035-2.

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210255: Computer Graphics Lab

| Lab Scheme: | Credit | Examination Scheme: |
|---|---|---|
| PR: 02 Hours/Week | 01 | TW: 25 Marks<br>PR: 50 Marks |

## Guidelines for Instructor's Manual

The instructor's manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface etc), University syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

## Guidelines for Student Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, conclusion/analysis. **Program codes with sample output of all performed assignments are to be submitted as softcopy.**

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

## Guidelines for Assessment

Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

## Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student's for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

## Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. Encourage students for the use of industry coding standards such as appropriate use of Hungarian notation, Indentation and comments. Use Display file where ever suitable.

Use of open source software is encouraged. In addition to these, instructor may assign one real life application in the form of a mini-project. Based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

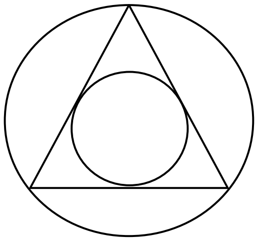
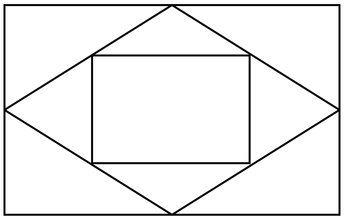
**Operating System recommended** :- 64-bit Open source Linux or its derivative

**Programming tools recommended**: - Open Source C++ Programming tool like G++/GCC.

Set of suggested assignment list is provided in groups- A, B, and C. Instructor is suggested to design lab assignments list by selecting/designing **12** suitable assignments- any 5 of group A, 5 from group B, **2** from group C (assignment number 26 is mandatory).

## Suggested List of Laboratory Assignments

### Group A

| | |
|---|---|
| 1. | Write C++/Java program to draw line using DDA and Bresenham's algorithm. Inherit pixel class and Use function overloading. |
| 2. | Write C++/Java program to draw circle using Bresenham's algorithm. Inherit pixel class. |
| 3. | Write C++/Java program to draw 2-D object and perform following basic transformations,<br>a) Scaling<br>b) Translation<br>c) Rotation<br>Use operator overloading. |
| 4. | Write C++/Java program to fill polygon using scan line algorithm. Use mouse interfacing to draw polygon. |
| 5. | A Mandelbrot Set is a set of complex number z that does not diverge under the transformation$x_{n+1} = x_n^2 + z$with $x_0 = 0$.Where, both x and z represent the complex numbers. Write C++/Java program to<br>a). Plot the Mandelbrot set for the threshold \|x\|= 2.<br>b) Plot Julia set choosing z ≠ 0. Use 254 colors for plotting in both cases. |
| 6. | Write C++/Java program to draw the polygons by using the mouse. Choose colors by clicking on the designed color pane. Use window port to draw. Use DDA algorithm for line drawing. |
| 7. | Write C++/Java program to draw inscribed and Circumscribed circles in the triangle as shown as an example below. (Use any Circle drawing and Line drawing algorithms)<br> |
| 8. | Write C++/Java program to draw the following pattern using any Line drawing algorithms.<br> |

| | |
|---|---|
| 9. | Write C++/Java program to draw a 4X4 chessboard rotated 45˚ with the horizontal axis. Use Bresenham algorithm to draw all the lines. Use seed fill algorithm to fill black squares of the rotated chessboard. |
| **Group B** | |
| 10. | Write C++/Java program for line drawing using DDA or Bresenhams algorithm with patterns such as solid, dotted, dashed, dash dot and thick. |
| 11. | Write C++/Java program to draw a convex polygon and fill it with desired color using Seed fill algorithm. Use mouse interfacing to draw polygon. |
| 12. | Write C++/Java program to draw a concave polygon and fill it with desired pattern using scan line algorithm. Use mouse interfacing to draw polygon. |
| 13. | Write C++/Java program to implement Cohen-Sutherland line clipping algorithm for given window. Draw line using mouse interfacing to draw polygon |
| 14. | Write C++/Java program to draw any object such as flower, waves using any curve generation techniques |
| 15. | Write C++/Java program to implement Painter's algorithm for hidden surface removal |
| 16. | Write C++/Java program to implement reflection of2-D object about X axis, Y axis and about X=Y axis. Also rotate object about arbitrary point given by user. |
| 17. | Write C++/Java program to generate Hilbert curve using concept of fractals. |
| 18. | Write C++/Java program to generate snowflake using concept of fractals. |
| 19. | Write C++/Java program to generate Bouncing ball animation using Direct3D/Maya/Blender |
| 20. | Write program to implement Cohen Sutherland Hodgman algorithm to clip any polygon. Provide the vertices of the polygon to be clipped and pattern of clipping interactively. |
| 21. | Write C++/Java program to implement translation, sheer, rotation and scaling transformations on equilateral triangle and rhombus. |
| **Group C** | |
| 22. | Write C++/Java program to draw 3-D cube and perform following transformations on it using OpenGL. a) Scaling b) Translation c) Rotation about one axis |
| 23. | Design and simulate any data structure like stack, queue, and trees using graphics. Simulation should include all operations performed on designed data structure. Implement the same using OpenGL. |
| 24. | Write C++/Java program to draw implement Cube rotation about vertical axis passing through its centroid. |
| 25. | Write C++/Java program to generate fractal patterns by using Koch curves. |
| 26. | Write C++/Java program to simulate any one of or similar scene-<br>• Clock with pendulum<br>• National Flag hoisting<br>• Vehicle/boat locomotion<br>• Water drop falling into the water and generated waves after impact<br>• Kaleidoscope views generation (at least 3 colorful patterns) |
| **Mini Project (Optional)-** Design and implement game / animation clip / Graphics Editor using open source graphics library. | |

# Savitribai Phule Pune University
## Second Year of Computer Engineering (2015 Course)
## 210256: Advanced Data Structures Lab

| Lab Scheme: | Credit | Examination Scheme: |
|---|---|---|
| PR: 04 Hours/Week | 02 | TW: 50 Marks<br>PR: 50 Marks |

### Guidelines for Instructor's Manual

The instructor’s manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/ preface etc), University syllabus, conduction & Assessment guidelines, topics under consideration-concept, objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

### Guidelines for Student Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and **handwritten write-up** of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept in brief, algorithm, flowchart, test cases, conclusion/analysis. **Program codes with sample output of all performed assignments are to be submitted as softcopy.**

As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

### Guidelines for Assessment

Continuous assessment of laboratory work is done based on overall performance and lab assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

### Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student’s for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

## Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. Encourage students for appropriate use of Hungarian notation, Indentation and comments. Use of open source software is encouraged.

In addition to these, instructor may assign one real life application in the form of a mini-project based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Set of suggested assignment list is provided in six groups. Each student must perform at least 13 assignments as at least 02 from group A, 02 from group B, 2 from group C, 2 from group D, 01 from group E, 01 from group F and 3 from group G.

**Operating System recommended** : 64-bit Open source Linux or its derivative
**Programming tools recommended**: Open Source C++ Programming tool like G++/GCC

## Suggested List of Laboratory Assignments

**Write C++/Java program for following-**

### Group A

| | |
|---|---|
| 1. | A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method. |
| 2 | Beginning with an empty binary search tree, Construct binary search tree by inserting the values in the order given. After constructing a binary tree -<br>i. Insert new node<br>ii. Find number of nodes in longest path<br>iii. Minimum data value found in the tree<br>iv. Change a tree so that the roles of the left and right pointers are swapped at every node<br>v. Search a value |
| 3 | For given expression eg. a-b*c-d/e+f construct inorder sequence and traverse it using postorder traversal(non recursive). |
| 4 | Read for the formulas in propositional calculus. Write a function that reads such a formula and creates its binary tree representation. What is the complexity of your function? |
| 5 | Given binary tree with n nodes, assign this tree to another [operator=] and then erase all nodes in a binary tree. |
| 6 | Convert given binary tree into threaded binary tree. Analyze time and space complexity of the algorithm. |
| 7 | Consider threading a binary tree using preorder threads rather than inorder threads. Design an algorithm for traversal without using stack and analyze its complexity. |
| 8 | A Dictionary stores keywords & its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Binary Search Tree for implementation. |

### Group B

| | |
|---|---|
| 9 | Write a function to get the number of vertices in an undirected graph and its edges. You may assume that no edge is input twice.<br>i. Use adjacency list representation of the graph and find runtime of the function<br>ii. Use adjacency matrix representation of the graph and find runtime of the function |
| 10 | There are flight paths between cities. If there is a flight between city A and city B then there is an edge between the cities. The cost of the edge can be the time that flight takes to reach city B from A, or the amount of fuel used for the journey. Represent this as a graph. The node can be represented by airport name or name of the city. Use adjacency list representation of the graph or use adjacency matrix representation of the graph. Justify the storage representation used. |
| 11 | You have a business with several offices; you want to lease phone lines to connect them up with each other; and the phone company charges different amounts of money to connect different pairs of cities. You want a set of lines that connects all your offices with a minimum total cost. Solve the problem by suggesting appropriate data structures. |
| 12 | Tour operator organizes guided bus trips across the Maharashtra. Tourists may have different preferences. Tour operator offers a choice from many different routes. Every day the bus moves from starting city S to another city F as chosen by client. On this way, the tourists can see the sights alongside the route travelled from S to F. Client may have preference to choose route. There is a restriction on the routes that the tourists may choose from, the bus has to take a short route from S to F or a route having one distance unit longer than the minimal distance. Two routes from S to F are considered different if there is at least one road from a city A to a city B which is part of one route, but not of the other route. |
| 13 | Consider the scheduling problem. n tasks to be scheduled on single processor. Let t1, ..., tn be durations required to execute on single processor is known. The tasks can be executed in any order but one task at a time. Design a greedy algorithm for this problem and find a schedule that minimizes the total time spent by all the tasks in the system. (The time spent by one is the sum of the waiting time of task and the time spent on its execution.) |
| **Group C** | |
| 14 | Consider telephone book database of N clients. Make use of a hash table implementation to quickly look up client's telephone number. |
| 15 | Implement all the functions of a dictionary (ADT) using hashing.<br>Data: Set of (key, value) pairs, Keys are mapped to values, Keys must be comparable, Keys must be unique<br>Standard Operations: Insert(key, value), Find(key), Delete(key) |
| 16 | For given set of elements create skip list. Find the element in the set that is closest to some given value. |
| 17 | The symbol table is generated by compiler. From this perspective, the symbol table is a set of name-attribute pairs. In a symbol table for a compiler, the name is an identifier, and the attributes might include an initial value and a list of lines that use the identifier.<br>Perform the following operations on symbol table:<br>(1) Determine if a particular name is in the table<br>(2) Retrieve the attributes of that name<br>(3) Modify the attributes of that name<br>(4) Insert a new name and its attributes<br>(5) Delete a name and its attributes |
| **Group D** | |

| | |
|---|---|
| 18 | Given sequence k = k1 <k2 < … < kn of n sorted keys, with a search probability pi for each key ki . Build the Binary search tree that has the least search cost given the access probability for each key. |
| 19 | A Dictionary stores keywords & its meanings. Provide facility for adding new keywords, deleting keywords, updating values of any entry. Provide facility to display whole data sorted in ascending/ Descending order. Also find how many maximum comparisons may require for finding any keyword. Use Height balance tree and find the complexity for finding a keyword |
| **Group E** | |
| 20 | To create ADT that implements the SET concept.<br>a. Add (newElement) -Place a value into the set b. Remove (element) Remove the value<br>c. Contains (element) Return true if element is in collection<br>d. Size () Return number of values in collection Iterator () Return an iterator used to loop over collection<br>e. Intersection of two sets, f. Union of two sets, g. Difference between two sets, h. Subset |
| 21 | Read the marks obtained by students of second year in an online examination of particular subject. Find out maximum and minimum marks obtained in that subject. Use heap data structure. Analyze the algorithm. |
| **Group F** | |
| 22 | Assume we have two input and two output tapes to perform the sorting. The internal memory can hold and sort m records at a time. Write a program in java for external sorting. Find out time complexity. |
| 23 | Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to main the data. |
| 24 | Company maintains employee information as employee ID, name, designation and salary. Allow user to add, delete information of employee. Display information of particular employee. If employee does not exist an appropriate message is displayed. If it is, then the system displays the employee details. Use index sequential file to maintain the data. |
| **Group G** | |
| 25 | Implement the Heap/Shell sort algorithm implemented in Java demonstrating heap/shell data structure with modularity of programming language. |
| 26 | Any application defining scope of Formal parameter, Global parameter, Local parameter accessing mechanism and also relevance to private, public and protected access. Write a Java program which demonstrates the scope rules of the programming mechanism. |
| 27 | Write a Java program which will demonstrate a concept of Interfaces and packages: In this assignment design and use of customized interfaces and packages for a specific application are expected. |
| 28 | Write a Java program which will demonstrate a concept of cohesion and coupling of the various modules in the program. |
| 29 | Write a program on template and exception handling in Java: in this assignment multiple templates are to be designed as a pattern and these patterns to be used to take decisions. |
| 30 | Write a Java program for the implementation of different data structures using JAVA collection libraries (Standard toolkit library): at least 5 data structures are used to design a suitable application. |
| 31 | Design a mini project using JAVA which will use the different data structure with or without Java collection library and show the use of specific data structure on the efficiency (performance) of the code. |

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210257: Microprocessor Lab

| Teaching Scheme: | Credit | Examination Scheme: |
|---|---|---|
| PR: 04 Hours/Week | 02 | TW: 25 Marks |
| | | PR: 50 Marks |

## Guidelines for Instructor's Manual

The instructor’s manual is to be developed as a hands-on resource and reference. The instructor's manual need to include prologue (about University/program/ institute/ department/foreword/preface), University syllabus, conduction & Assessment guidelines, topics under consideration concept objectives, outcomes, set of typical applications/assignments/ guidelines, and references.

## Guidelines for Student's Lab Journal

The laboratory assignments are to be submitted by student in the form of journal. Journal consists of prologue, Certificate, table of contents, and write-up of each assignment (Title, Objectives, Problem Statement, Outcomes, software & Hardware requirements, Date of Completion, Assessment grade/marks and assessor's sign, Theory- Concept, instructions/features used, test cases, conclusion/analysis and references).

Program codes with sample output of all performed assignments are to be submitted as softcopy. As a conscious effort and little contribution towards Green IT and environment awareness, attaching printed papers as part of write-ups and program listing to journal may be avoided. Use of DVD containing students programs maintained by lab In-charge is highly encouraged. For reference one or two journals may be maintained with program prints at Laboratory.

## Guidelines for Lab /TW Assessment

Continuous assessment of laboratory work is based on overall performance and lab assignments performance of student. Each lab assignment assessment will assign grade/marks based on parameters with appropriate weightage. Suggested parameters for overall assessment as well as each lab assignment assessment include- timely completion, performance, innovation, efficient codes, punctuality and neatness.

## Guidelines for Practical Examination

Both internal and external examiners should jointly set problem statements. During practical assessment, the expert evaluator should give the maximum weightage to the satisfactory implementation of the problem statement. The supplementary and relevant questions may be asked at the time of evaluation to test the student for advanced learning, understanding of the fundamentals, effective and efficient implementation. So encouraging efforts, transparent evaluation and fair approach of the evaluator will not create any uncertainty or doubt in the minds of the students. So adhering to these principles will consummate our team efforts to the promising start of the student's academics.

## Guidelines for Laboratory Conduction

The instructor is expected to frame the assignments by understanding the prerequisites, technological aspects, utility and recent trends related to the topic. The assignment framing policy need to address the average students and inclusive of an element to attract and promote the intelligent students. The instructor may set multiple sets of assignments and distribute among batches of students. It is appreciated if the assignments are based on real world problems/applications. Use of open source software is encouraged.

In addition to these, instructor may assign one real life application in the form of a mini-project based on the concepts learned. Instructor may also set one assignment or mini-project that is suitable to respective branch beyond the scope of syllabus.

Operating System: Latest 64-bit Version and update of Microsoft Windows 7/ Windows 8 Operating System onwards or 64-bit Open source Linux or its derivative.

Programming Tools: Preferably using Linux equivalent or MASM 64x or equivalent, Microsoft Visual Studio x64 Intrinsic.

### Suggested List of Laboratory Assignments (Any 12)

| | |
|---|---|
| 1. | Write X86/64 ALP to count number of positive and negative numbers from the array |
| 2. | Write X86/64 ALP to perform non-overlapped and overlapped block transfer (with and without string specific instructions). Block containing data can be defined in the data segment. |
| 3. | Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for:<br>(a) HEX to BCD b) BCD to HEX (c) EXIT.<br>Display proper strings to prompt the user while accepting the input and displaying the result. (wherever necessary, use 64-bit registers) |
| 4. | Write X86/64 ALP to perform multiplication of two 8-bit hexadecimal numbers. Use successive addition and add and shift method. (use of 64-bit registers is expected) |
| 5. | Write X86 ALP to find, a) Number of Blank spaces b) Number of lines c) Occurrence of a particular character. Accept the data from the text file. The text file has to be accessed during Program_1 execution and write FAR PROCEDURES in Program_2 for the rest of the processing. Use of PUBLIC and EXTERN directives is mandatory. |
| 6. | Write X86/64 ALP to switch from real mode to protected mode and display the values of GDTR, LDTR, IDTR, TR and MSW Registers. |
| 7. | Write X86 program to sort the list of integers in ascending/descending order. Read the input from the text file and write the sorted data back to the same text file using bubble sort |
| 8. | Write X86 menu driven Assembly Language Program (ALP) to implement OS (DOS) commands TYPE, COPY and DELETE using file operations. User is supposed to provide command line arguments in all cases. |
| 9. | Write x86 ALP to find the factorial of a given integer number on a command line by using recursion. Explicit stack manipulation is expected in the code. |
| 10. | Write 80387 ALP to find the roots of the quadratic equation. All the possible cases must be considered in calculating the roots. |
| 11. | Write 80387 ALP to plot Sine Wave, Cosine Wave and Sinc function. Access video memory directly for plotting. |
| 12. | Write 80387 ALP to obtain: i) Mean ii) Variance iii) Standard Deviation Also plot the histogram for the data set. The data elements are available in a text file. |
| 13. | Write a Terminate but Stay Resident (TSR) program for a key-logger. The key-presses during the stipulated time need to be displayed at the center of the screen. **OR**<br>Write a TSR to generate the pattern of the frequency tones by reading the Real Time Clock (RTC). The duration of the each tone is solely decided by the programmer. |
| 14. | Write 80386 ALP to implement multitasking. Where each task is supposed to change the color of the text displayed at the center of the screen |

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210258: Audit Course 2

In addition to credits, it is recommended that there should be audit course in preferably in each semester from second year to supplement their knowledge and skills. Student will be awarded the bachelor's degree if he/she earns 190 credits and clears all the audit courses specified in the syllabus. The student will be awarded grade as AP on successful completion of audit course.

The student may opt for one of the audit courses per semester, starting in second year first semester. Though not mandatory, such a selection of the audit courses helps the learner to explore the subject of interest in greater detail resulting in achieving the very objective of audit course's inclusion. List of options offered is provided. Each student has to choose one audit course from the list per semester. Evaluation of audit course will be done at institute level itself. Method of conduction and method of assessment for audit courses are suggested.

## Criteria:

The student registered for audit course shall be awarded the grade AP(Audit Course Pass) and shall be included such AP grade in the Semester grade report for that course, provided student has the minimum attendance as prescribed by the Savitribai Phule Pune University and satisfactory in-semester performance and secured a passing grade in that audit course. No grade points are associated with this 'AP' grade and performance in these courses is not accounted in the calculation of the performance indices SGPA and CGPA. Evaluation of audit course will be done at institute level itself. (Ref- http://www.unipune.ac.in/Syllabi_PDF/revised-2015/engineering/UG_RULE_REGULATIONS_FOR_CREDIT_SYSTEM-2015_18June.pdf)

## Guidelines for Conduction and Assessment (Any one or more of following but not limited to)

- Lectures/ Guest Lectures
- Visits (Social/Field) and reports
- Demonstrations
- Surveys
- Mini Project
- Hands on experience on specific focused topic

## Guidelines for Assessment (Any one or more of following but not limited to)

- Written Test
- Demonstrations/ Practical Test
- Presentations
- IPR/Publication
- Report

## Audit Course 2 Options

| Audit Course Code | Audit Course Title |
|---|---|
| AC2-I | Water Management |
| AC2-II | Intellectual Property Rights and Patents |
| AC2-III | The Science of Happiness |
| AC2-IV | Stress Relief: Yoga and Meditation |
| AC2-V | Foreign Language (one of Japanese/Spanish/French/German) Course contents for **Japanese( Module 2)** are provided. For other languages institute may design suitably. |

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210258: Audit Course 2
# AC2-I: Water Management

Water is a vital resource for all life on the planet. Only three percent of the water resources on Earth are fresh and two-thirds of the freshwater is locked up in ice caps and glaciers. One fifth of the remaining one percent is in remote, inaccessible areas. As time advances, water is becoming scarcer and having access to clean, safe, drinking water is limited among countries. Pure water supply and disinfected water treatment are prerequisites for the well-being of communities all over the world. One of the biggest concerns for our water-based resources in the future is the sustainability of the current and even future water resource allocation. This course will provide students a unique opportunity to study water management activities like planning, developing, distributing and optimum use of water resources. This course covers the topics that management of water treatment of drinking water, industrial water, sewage or wastewater, management of water resources, management of flood protection.

## Course Objectives:

- To develop understanding of water recourses.
- To study global water cycle and factors that affect this cycle.
- To analyze the process for water resources and management.
- To study the research and development areas necessary for efficient utilization and management of water recourses.

## Course Outcomes:

On completion of the course, learner will be able to–

- Understanding of the global water cycle and its various processes
- Understanding of climate change and their effects on water systems
- Understanding of Drinking treatment and quality of groundwater and surface water
- Understanding of the Physical, chemical, and biological processes involved in water treatment and distribution.

## Course Contents:

1. **Understanding ‘water’-Climate change and the global water cycle, Understanding global hydrology**
2. Water resources planning and management-Water law and the search for sustainability: a comparative analysis, Risk and uncertainty in water resources planning and management
3. Agricultural water use -The role of research and development for agriculture water use
4. Urban water supply and management - The urban water challenge, Water sensitive urban design

## References:

1. R. Quentin Graft, Karen Hussey, Quentin Graft, Karen Hussey, Publisher, "Water Resources Planning and Management”, Cambridge University Press, ISBN: 9780511974304, 9780521762588.
2. P. C. Basil, "Water Management in India”, ISBN: 8180690970, 2004.
3. C.A. Brebbia, "Water Resources Management”, ISBN: 978-1-84564-960-9

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210258: Audit Course 2
# AC2-II: Intellectual Property Rights and Patents

Intellectual property is the area of law that deals with protecting the rights of those who create original works. It covers everything from original plays and novels to inventions and company identification marks. The purpose of intellectual property laws is to encourage new technologies, artistic expressions and inventions while promoting economic growth.

Innovation and originality have great potential value. Whatever line of activity you are engaged in, future success depends on them. The last few years have seen intellectual property rights become an issue of general interest: the smart phone "patent wars", the introduction of Digital Rights management (DRM) and the rise of generic pharmaceuticals and open-source software are just some examples that have been in the public eye. Protecting your intellectual rights appropriately should be a top priority. Yet too many people embark on their chosen professions without even a basic awareness of intellectual property.

## Course Objectives:

- To encourage research, scholarship, and a spirit of inquiry
- To encourage students at all levels to develop patentable technologies.
- To provide environment to the students of the Institute for creation, protection, and commercialization of intellectual property and to stimulate innovation.

## Course Outcomes:

On completion of the course, learner will be able to–

- Understand the fundamental legal principles related to confidential information, copyright, patents, designs, trademarks and unfair competition
- Identify, apply and assess principles of law relating to each of these areas of intellectual property
- Apply the appropriate ownership rules to intellectual property you have been involved in creating

## Course Contents:

- **Introduction to Intellectual Property Law** – The Evolutionary Past - The IPR Tool Kit- Para -Legal Tasks in Intellectual Property Law
- **Introduction to Trade mark** – Trade mark Registration Process – Post registration Procedures – Trade mark maintenance - Transfer of Rights – Inter partes Proceeding – Infringement - Dilution Ownership of Trade mark
- **Introduction to Copyrights** – – Principles of Copyright Principles -The subjects Matter of Copy right – The Rights Afforded by Copyright Law – Copy right Ownership, Transfer and duration – Right to prepare Derivative works
- **Introduction to Trade Secret** – Maintaining Trade Secret – Physical Security – Employee Limitation - Employee confidentiality agreement

## References:

1. Debirag E. Bouchoux: "Intellectual Property". Cengage learning ISBN-10:1111648573
2. Ferrera, Bird, Darrow, "Cyber Law. Texts & Cases", South- ISBN:0-324-39972-3
3. Prabhuddha Ganguli: "Intellectual Property Rights" TMH, ISBN-10:0070077177

# Savitribai Phule Pune University
## Second Year of Computer Engineering (2015 Course)
## 210258: Audit Course 2
## AC2-III : The Science of Happiness

Everybody wants to be happy. One can explore innumerable ideas about what happiness is and how we can get some. But not many of those ideas are based on science. That's where this course comes in. The "The subject "Science of Happiness" aims to teach the pioneering science of positive psychology, which explores the ancestry of a happy and meaningful life. Clinical psychologists have been dealing with miserable feelings since their discipline was established. In the last 30 years, neuroscientists have made major headway in the understanding of the sources of anger, depression, and fear.

Today, whole industries profit from this knowledge—producing pills for every sort of pathological mood disturbance. But until recently, few neuroscientists focused on the subject of happiness. This course focuses on discovering how cutting-edge research can be applied to their lives. Students will learn about the Intra-disciplinary research supporting this view, spanning the fields of psychology, neuroscience, evolutionary biology, and beyond. The course offers students practical strategies for tapping into and nurturing their own happiness, including trying several research-backed activities that foster social and emotional well-being, and exploring how their own happiness changes along the way.

### Course Objectives:

- To understand the feeling of happiness
- To study the sources of positive feelings
- To analyze the anatomy of the happiness system
- To study the effect of thoughts and emotions on the happiness system

### Course Outcomes:

On completion of the course, learner will be able to–

- Ability to understand what happiness is and why it matters to you
- Ability to learn how to increase your own happiness
- Understanding of the power of social connections and the science of empathy
- Ability to understand what is mindfulness and its real world applications

### Course Contents:

1. Happiness: what is it?
2. The secret of smiling
3. The autonomy of positive feelings
4. Positive feelings as a compass
5. The happiness system
6. Foundations: Emotions, Motivation and nature of Well being
7. Subjective well being
8. Love and well being
9. Optimal well being
10. Religion, Spirituality and well being

### References:

1. Happier, Stefan Klein , "The Science of Happiness, How Our Brains Make Us Happy and what We Can Do to Get", Da Capo Press, ISBN 10: 156924328X, 13: 978-1569243282.
2. C. Compton, Edward Hoffman, "Positive Psychology: The Science of Happiness and Flourishing", William, Cengage Learning, 2012, ISBN10: 1111834121.

# Savitribai Phule Pune University
# Second Year of Computer Engineering (2015 Course)
# 210258: Audit Course 2
# AC2-IV: Stress Relief: Yoga and Meditation

The concepts and practices of Yoga originated in India about several thousand years ago. Its founders were great Saints and Sages. The great Yogis presented rational interpretation of their experiences of Yoga and brought about a practical and scientifically sound method within every one's reach. Yoga today, is no longer restricted to hermits, saints, and sages; it has entered into our everyday lives and has aroused a worldwide awakening and acceptance in the last few decades. The science of Yoga and its techniques have now been reoriented to suit modern sociological needs and lifestyles.

Yoga is one of the six systems of Vedic philosophy. The Yoga advocates certain restraints and observances, physical discipline, breathe regulations, restraining the sense organs, contemplation, meditation and Samadhi. The practice of Yoga prevents psychosomatic disorders and improves an individual's resistance and ability to endure stressful situations.

## Course Objectives:

- To impart knowledge about the basic technique and practice of yoga, including instruction in breath control, meditation, and physical postures
- To gain an intellectual and theoretical understanding of the principles embodied in the Yoga Sutras, the Bhagavad-Gita, and other important texts and doctrines
- Relaxation and stress reduction ,Personal insight and self understanding, Personal empowerment, Gaining wisdom and spiritual discernment
- Awakening the abilities or powers of the Super conscious mind

## Course Outcomes:

On completion of the course, learner will be able to–

- Students understanding of philosophy and religion as well as daily life issues will be challenged and enhanced.
- Enhances the immune system.
- Intellectual and philosophical understanding of the theory of yoga and basic related Hindu scriptures will be developed.
- Powers of concentration, focus, and awareness will be heightened.

## Course Contents:

1. Meaning and definition of yoga – Scope of Yoga - Aims and Objectives of Yoga – Misconception about yoga.
2. Ayurveda: an introduction to this system of health care derived from the Vedic tradition Anatomy and Physiology as they relate to Yoga
3. Yoga Philosophy and Psychology

## References:

1. B.K.S. Iyengar, "BKS Iyengar Yoga The Path to Holistic Health", DK publisher, ISBN-13: 978-1409343479
2. Osho, "The Essence of Yoga", Osho International Foundation, ISBN: 9780918963093

# Savitribai Phule Pune University, Pune
# Second Year of Computer Engineering (2015 Course)
# 210258: Audit Course 2
## AC2-V: Foreign Language ( Japanese) Module 2

With changing times, the competitiveness has gotten into the nerves and ‘Being the Best’ at all times is only the proof of it. Nonetheless, ‘being the best’ differs significantly from ‘Communicating the best’. The best can merely be communicated whilst using the best suited Language!!

## Course Objectives:

- To meet the needs of ever growing industry with respect to language support.
- To get introduced to Japanese society and culture through language.

## Course Outcomes:

On completion of the course student-

- will have ability of basic communication.
- will have the knowledge of Japanese script.
- will get introduced to reading, writing and listening skills for language Japanese.
- will develop interest to pursue professional Japanese Language course.

## Course Contents:

- Katakana basic Script, Denoting things ( nominal & prenominal demonstratives ), Purchasing at the Market / in a shop / mall (asking & stating price)

- Katakana : Modified kana, double consonant, letters with ya, yu, yo, Long vowels, Describing time, describing starting & finishing time ( kara ~ made ), Point in time (denoting the time when any action or the movement occurs)

- Means of transport (Vehicles), Places, Countries, Stating Birth date, Indicating movement to a certain place by a vehicle

## References:

1. Minna No Nihongo, “Japanese for Everyone”, (Indian Edition), Goyal Publishers & Distributors Pvt. Ltd.
2. http://www.tcs.com (http://www.tcs.com/news_events/press_releases/Pages/TCS-Inaugurates-Japan-centric-Delivery-Center-Pune.aspx)