

**Savitribai Phule Pune University**

**Three Year Degree Course in  
B. Sc. Computer Science**

## 1) Title of the Course : B. Sc. Computer Science

### T. Y. B. Sc. Computer Science Syllabus in the Subject Computer Science (To be implemented from Academic Year 2014-15)

#### 2) Preamble:

B. Sc. Computer Science is a systematically designed three year course that prepares the student for a career in Software Industry. The syllabus of computer Science subject along with that of the three allied subjects (Mathematics, Electronics and Statistics) forms the required basics for pursuing higher studies in Computer Science. The Syllabus also develops requisite professional skills and problem solving abilities for pursuing a career in Software Industry.

#### 3) Introduction:

At **first year of under-graduation** basic foundation of two important skills required for software development is laid. A course in programming and a course in database fundamentals forms the preliminary skill set for solving computational problems. Simultaneously two practical courses are designed to supplement the theoretical training. The second practical course also includes a preliminary preparation for website designing in the form of HTML programming.

Alongwith Computer Science two theory and one practical course each in Statistics, Mathematics and Electronics help in building a strong foundation.

At **second year under-graduation**: The programming skills are further strengthened by a course in Data structures and Object oriented programming. The advanced topics in Databases and preliminary software engineering form the second course. Two practical courses alongside help in hands-on training. Students also undertake a mini project using software engineering principles to solve a real world problem. Simultaneously two theory and one practical course each in Mathematics and Electronics help in strengthening problem solving abilities.

At **third year under-graduation**: Six theory papers in each semester and practical courses cover the entire spectrum of topics necessary to build knowledge base and requisite skill set. Third practical course also includes project work which gives students hands on experience in solving a real world problem.

#### Objectives:

- To develop problem solving abilities using a computer
- To build the necessary skill set and analytical abilities for developing computer based solutions for real life problems.
- To imbibe quality software development practices.
- To create awareness about process and product standards
- To train students in professional skills related to Software Industry.
- To prepare necessary knowledge base for research and development in Computer Science
- To help students build-up a successful career in Computer Science

#### 4) Eligibility:

Higher Secondary School Certificate (10+2) Science stream or its equivalent Examination as per the University of Pune eligibility norms.

**Note:** Admissions will be given as per the selection procedure / policies adopted by the respective college, in accordance with conditions laid down by the University of Pune. Reservation and relaxation will be as per the Government rules.

#### 5 A) Examination Pattern:

## First Year B. Sc. Computer Science

### Subject : Computer Science

Pattern of Examination: Annual for both Theory and Practical Courses

Paper/ Course No.	Title	Total Number of lectures/practicals per Term	Standard of passing		
			Internal marks out of 20	External marks out of 80	Total marks out of 100
Computer Science Paper I (CS-101)	Problem Solving Using Computers and 'C' Programmin g	Three lectures/Week (Total 80 lectures )	08	32	40 *
Computer Science Paper II CS-102)	File Organizatio n and Fundament al of Databases	Three lectures/Week (Total 80 lectures )	08	32	40 *
Computer Science Practical Paper I (CS-103)	Computer Science Practical Paper I	25 Practical slots of 4 lectures each	08	32	40 *
Computer Science Practical Paper II (CS-104)	Computer Science Practical Paper II	25 Practical slots of 4 lectures each	08	32	40 *

\* Subject to compulsory passing in external examination and getting minimum 40 marks out of 100

Notes:

1. Total marks: Theory (100 + 100 ) = 200 marks
2. Total marks per year 200 (Theory) + 100 marks (practical)+ Grade(practical) = 300 marks +Grade
3. Internal marks for theory papers given on the basis of internal assessment tests and for practicals on continuous assessment of lab work.
4. In case of Computer Science Practical Paper II, marks out of 100 will be converted to grades

Marks	Grade
75 And Above	O
65 And Above	A
55 and above	B
50 And above	C
45 And Above	D

40 And Above	E
Below 40 ( indicates Failure)	F

**Theory examination** will be of three hours duration for each theory course. There shall be 5 questions each carrying equal marks. The pattern of question papers shall be:

Question 1	8 sub-questions, each of 2 marks; answerable in 2 -3 lines and based on entire syllabus
Question 2, 3 ,4 and 5	4 out of 5/6– short answer type questions; answerable in 8 – 10 lines mix of theory and problems

**Internal examination:** Internal assessment of the student by respective teacher will be based on written test, 10 marks each term. The written test shall comprise of objective type questions – Multiple Type Questions, True / False, Definitions, Answer in Two or three line question (Describe/Explain). There shall be 20 questions. Practical: Continuous assessment of Lab work and mini project.

**Practical Examination:** Practical examination shall be conducted by the respective college at the end of the academic year. Practical examination will be of 3 hours duration for each practical course. Certified journal is compulsory to appear for practical examination. There shall be two expert and two examiners per batch for the practical examination.

**Second Year B. Sc. (Computer Science)**  
**Subject : Computer Science**

No	Paper	Title: Semester I	Title: Semester II
1	Computer Science Paper I	CS-211:Data Structures using 'C'	CS-221:Object Oriented Concepts using C++
2	Computer Science Paper II	CS-212: Relational Database Management System	CS-222:Software Engineering
3	Computer Science Paper III	CS-223:Data structures Practicals and C++ Practicals	
4	Computer Science Paper IV	CS-224:Database Practicals & Mini Project using Software Engineering techniques	

**Pattern of examination: Semester**

Theory courses (Sem I: CS-211 and CS212): Semester  
(Sem II: CS-221 and CS-222): Semester  
Practical Course (CS-223 and CS-224): Annual

Paper/Course No.	Title	Total Number of Lectures/Practicals Per Week	Standard Of Passing		
			Internal marks out of 10 (theory) Out of 20 (practicals)	External marks out of 40 (theory) Out of 80 (practicals)	Total passing marks out of 50 (theory) and out of 100 (practicals)

Theory Paper I (CS-211)	Data Structures using 'C'	Four Lectures/per Week (Total 48 per Semester)	04	16	20*
Theory Paper II (CS-212)	Relational Database Management System	Four Lectures/per Week (Total 48 per Semester)	04	16	20*
Theory Paper I (CS-221)	Object Oriented Concepts using C++	Four Lectures/per Week (Total 48 per Semester)	04	16	20*
Theory Paper II (CS-222)	Software Engineering	Four Lectures/per Week (Total 48 per Semester)	04	16	20*
Practical paper I (CS 223) (First & Second Semester)	Data structures Practicals and C++ Practicals	Practicals of 4 lectures each 25 practicals / year)	08	32	40*
Practical paper II (CS 224) (First & Second Semester)	Database Practicals & Mini Project using Software Engineering techniques	Practicals of 4 lectures each 25 practicals / year)	08	32	40**

\* Subject to compulsory passing in external examination and getting minimum 20 marks out of 50

\*\* Subject to compulsory passing in external examination and getting minimum 40 marks out of 100

Notes:

1. Total marks: Theory for each semester (50 + 50 ) = 100 marks
2. Total marks per year 200 (Theory) + 100 marks (practicals)+Grade(practical) = 300 marks+Grade
3. Internal marks for theory papers given on the basis of Continuous internal Assessment

**Theory examination** will be of two hours duration for each theory course. There shall be 4 questions carrying equal marks. The pattern of question papers shall be:

Question 1	10 sub-questions, each of 1 mark; answerable in 2 -3 lines and based on entire syllabus	10 Marks
Question 2, 3	Sub-questions carrying 5 marks (2 out of 3)	10 Marks
Question 4	Sub-questions carrying marks depending on their	10 Marks

	complexity with options	
--	-------------------------	--

**Internal examination:** Internal assessment of the student by respective teacher will be based on written test, 10 marks each Semester. The written test shall comprise of objective type questions – Multiple Type Questions, True / False, Definitions, Answer in Two or three line question (Describe/Explain) There shall be 20 questions.

Practicals: Continuous assessment of practical performance

**Practical Examination:** Practical examination shall be conducted at the respective college at the end of the academic year. Practical examination will be of 3 hours duration. Continuous assessment of practical performance should be using a Lab Book specifically designed for the purpose. Certified Lab book is compulsory to appear for practical examination. There is no need of attaching program printouts to the Lab Book. There shall be one expert and two examiners per batch for the practical examination. One of the examiners will be external.

### Third Year B. Sc. (Computer Science)

No	Paper	Title: Semester I	Title: Semester II
1	Computer Science Paper I	CS-331: System Programming	CS-341: Operating System
2	Computer Science Paper II	CS-332: Theoretical Computer Science	CS-342: Compiler Construction
3	Computer Science Paper III	CS-333: Computer Networks-I	CS-343: Computer Networks-II
4	Computer Science Paper IV	CS-334: Internet Programming- I	CS-344: Internet Programming- II
5	Computer Science Paper V	CS-335: Programming in Java-I	CS-345: Programming in Java-II
6	Computer Science Paper VI	CS-336: Object Oriented Software Engineering	CS-346: Computer Graphics
7	Computer Science Paper VII	CS-347: Practicals Based on CS-331 and CS341 – Sem I & Sem II	
8	Computer Science Paper VIII	CS-348: Practicals Based on CS-335 and CS-344 – Sem I & Sem II and Computer Graphics using Java	
9	Computer Science Paper IX	CS-349: Practicals Based on CS-334 and CS-344 – Sem I & Sem II and Project	

Pattern of examination: Semester

Theory courses:

(Sem III: CS-331-CS-336): Semester (Sem IV: CS-341-CS-346): Semester

Practical Course:

(CS-347-CS-349): Annual

Theory Papers					
Paper/Course No.	Title	Total Number of Lectures/Practicals Per Week	Standard Of Passing		
			Internal marks out of 10	External marks out of 40	Total passing marks out

			(theory) Out of 20 (practicals)	(theory) Out of 80 (practicals)	of 50 (theory) and out of 100 (practicals)
<b>SEM III</b>					
Theory Paper I (CS-331)	System Program min g	48	04	16	20*
Theory Paper II (CS-332)	Theoretica l Computer Science	48	04	16	20*
Theory Paper III (CS-333)	Computer Networks- I	48	04	16	20*
Theory Paper IV (CS-334)	Internet Programm ing I	48	04	16	20*
Theory Paper V (CS-335)	Program min g in Java- I	48	04	16	20*
Theory Paper V (CS-336)	Object Oriented Software Engineeri ng	48	04	16	20*
<b>SEM IV</b>					
Theory Paper I (CS-341)	Operating System	48	04	16	20*
Theory Paper II (CS-342)	Compiler Constructi on	48	04	16	20*
Theory Paper III (CS-343)	Computer Networks- II	48	04	16	20*
Theory Paper IV (CS-344)	Internet Programm ing II	48	04	16	20*
Theory Paper V (CS-345)	Program min g in Java- II	48	04	16	20*
Theory Paper V (CS-346)	Computer Graphics	48	04	16	20*
<b>Practical Papers</b>					
Practical paper I	Practicals	Practicals of 4			

CS 347 (Semester III & IV)	Based on CS-331 and CS-341 – Sem I & Sem II	lectures each 25 practicals / year)	08	32	40**
Practical paper II CS 348 (Semester III & IV)	CS-348:Practicals Based on CS-335 and CS-345 – Sem I & Sem II and Computer Graphics using OpenGL	Practicals of 4 lectures each 25 practicals / year)	08	32	40**
Practical paper I CS 349 (Semester III & IV)	CS-349:Practicals Based on CS-334 and CS-344 – Sem I & Sem II and Project	Practicals of 4 lectures each 25 practicals / year)	08	32	40**

\* Subject to compulsory passing in external examination and getting minimum 20 marks out of 50

\*\* Subject to compulsory passing in external examination and getting minimum 40 marks out of 100

Notes:

1. Total marks: Theory for each semester (50 \_ 6 ) = 300 marks
2. Total marks per year 600 (Theory) + 300 marks (practicals) = 900 marks
3. Internal marks for theory papers given on the basis of continuous internal assessment

**Theory examination** will be of two hours duration for each theory course. There shall be 4 questions carrying equal marks. The pattern of question papers shall be:

**Theory examination** will be of two hours duration for each theory course. There shall be 4 questions carrying equal marks. The pattern of question papers shall be:

Question 1	10 sub-questions, each of 1 mark; answerable in 2 -3 lines and based on entire syllabus	10 Marks
Question 2, 3	Sub-questions carrying 5 marks (2 out of 3)	10 Marks
Question 4	Sub-questions carrying marks depending on their	10 Marks

	complexity with options	
--	-------------------------	--

**Internal examination:** Internal assessment of the student by respective teacher will be based on written test, 10 marks each Semester. The written test shall comprise of objective type questions – Multiple Type Questions, True / False, Definitions, Answer in Two or three line question (Describe/Explain) There shall be 20 questions.

Practicals: Continuous assessment of practical performance

**Practical Examination:** Practical examination shall be conducted at the respective college at the end of the academic year. Practical examination will be of 3 hours duration. Continuous assessment of practical performance should be using a Lab Book specifically designed for the purpose. Certified Lab book is compulsory to appear for practical examination. There shall be one expert and two examiners per batch for the practical examination. One of the examiners will be external.

**5 B) Standard of Passing:**

i. In order to pass in the first year theory examination, the candidate has to obtain 40 marks out of 100 in each course. (Minimum 32 marks out of 80 must be obtained in the University Theory Examination.)

ii. In order to pass in the Second Year and Third Year theory examination, the candidate has to obtain 20 marks out of 50 in each course of each semester.

**5 C) ATKT Rules:**

While going from F.Y.B.Sc. to S.Y.B.Sc. at least 8 courses (out of total 13) should be passed; however all F.Y.B.Sc. courses should be passed while going to T.Y.B.Sc. While going from S.Y.B.Sc. to T.Y.B.Sc., at least 12 courses (out of 22) should be passed (Practical Course at S.Y.B.Sc. will be equivalent to 2 courses).

**5 D) Award of Class:**

The class will be awarded to the student on the aggregate marks obtained during the second and third year in the principal subject only. The award of the class shall be as follows:

1	Aggregate 70% and above	First Class with Distinction
2	Aggregate 60% and more but less than 70%	First Class
3	Aggregate 55% and more but less than 60%	Higher Second Class
4	Aggregate 50% and more but less than 55%	Second Class
5	Aggregate 40% and more but less than 50%	Pass Class
6	Below 40%	Fail

**5 E) External Students:** There shall be no external students.

**5 F) Setting question papers:**

**F.Y.B.Sc.:** For theory papers I and II annual question papers shall be set by the University of Pune and assessment done at the respective colleges. Questions should be designed to test the conceptual knowledge and understanding of the basic concepts of the subject. For Practical Papers, the Question paper slips will be provided by the University of Pune and assessment done at the respective colleges.

**S.Y.B.Sc. and T.Y.B.Sc.:**For theory papers I and II for each semester and also for the annual practical examination question papers set by the University of Pune.

Centralized assessment for theory papers done as per the University instructions.

Questions should be designed to test the conceptual knowledge and understanding of the basic concepts of the subject. For Practical Papers: Papers shall be set by the University of Pune and assessment done by the internal examiner and external examiner appointed by University of Pune.

**5G) Verification and Revaluation Rules:**

As per university Statues and rules for verification and revaluation of marks in stipulated time after declaration of the semester examination result.

**6) Course Structure:**

**Duration:** The duration of B.Sc. Computer Science Degree Program shall be three years.

**a) All are Compulsory Papers:**

F.Y.B.Sc. : 2 Theory + 2 Practical (Annual)

S .Y.B.Sc.: 2 Theory per semester + 2 Practical (Annual)

T.Y.B.Sc.: 6 Theory per semester + 3 Practical (Annual)

**b) Question Papers :**

**F.Y.B.Sc. Theory paper:**

University Examination – 80 marks (at the end of 2nd term)

Internal Examination – 20 marks

**S.Y / T.Y. - B.Sc.Theory paper:**

University Examination – 40 marks (at the end of each term)

Internal Examination – 10 marks

**F.Y. / S.Y / T.Y. - B.Sc. Practical Paper:**

University Examination – 80 marks (at the end of 2nd term)

Internal Examination – 20 marks

**c) Medium of Instruction:** The medium of instruction for the course shall be **English.**

**7) Equivalence of Previous Syllabus:**

<b>Old Course (2008 Pattern)</b>	<b>New Course (2013 Pattern)</b>
CS 331: System Programming & Operating System I	CS 331 : System Programming
CS 341: System Programming & Operating System II	CS 341 : Operating System
CS 332 : Theoretical Computer Science & Compiler Construction I	CS 332 : Theoretical Computer Science
CS 342 : Theoretical Computer Science & Compiler Construction II	CS 342 : Compiler Construction
CS 333 :Computer Networks I	CS 333 :Computer Networks I
CS 343 :Computer Networks II	CS 343 :Computer Networks II
CS 334 :Web development and PHP programming I	CS 334 :Internet Programming I
CS 344 : Web development and PHP programming II	CS 344 :Internet Programming II
CS 335 :Programming in Java I	CS 335 :Programming in Java I
CS 345 :Programming in Java II	CS 345 :Programming in Java II
CS 336 :Object Oriented Software Engineering	CS 336 :Object Oriented Software Engineering
CS 346 :Business Applications	CS 346 :Computer Graphics
CS 347: Lab Course I	CS 347: Lab Course I
CS 348:Lab Course II	CS 348:Lab Course II
CS 349: Lab Course III	CS 349: Lab Course III

**8) University Terms:** Dates for commencement and conclusion for the first and second terms will be declared by the University authorities. Terms can be kept by only duly admitted students. The term shall be granted only on minimum 75 percent attendance at theory and practical course and satisfactory performance during the term.

**9) Qualification of Teachers:** M.Sc. Computer Science/M.C.A. or equivalent master degree in science with class/grades and NET/SET as per prevailing University/Government /UGC rules.

**10) Detail Syllabus with Recommended Books:**

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Systems Programming**  
**Code No. : CS-331**

---

---

**Semester III**

**Total Lectures : 48**

**Aim :** To understand the design and implementation issues of System programs that play an important role in program development.

**Objectives :**

- To understand the design structure of a simple editor.
- To understand the design structure of Assembler and macro processor for an hypothetical simulated computer.
- To understand the working of linkers and loaders and other development utilities.
- To understand Complexity of Operating system as a software.

**1. Introduction**

**[4]**

- 1.1. Types of program – System program and Application program.
- 1.2. Difference between system programming and application programming.
- 1.3. Elements of Programming environment - Editor, Preprocessor, Assembler, Compiler, Interpreter, Linker and Loader, Debugger, Device drivers, Operating System.
- 1.4. Simulation of simple computer smac0 (hypothetical computer) -Memory, Registers, Condition Codes, Instruction format, Instruction Set, smac0 programs.

**2. Editors**

**[2]**

- 2.1 Definition, need/purpose of editor.
- 2.2 Types of editor- Examples ed, sed, VIM & emacs
- 2.3 Structure of editor

**3. Assembler**

**[10]**

- 3.1 Definition.
- 3.2 Features of assembly language, advantages
- 3.3 Statement format, types of statements – Imperative, Declarative, Assembler Directive.
- 3.4 Constants and Literals.
- 3.5 Advanced assembler directives (LTORG, ORIGIN, EQU),
- 3.6 Design of assembler – Analysis Phase and Synthesis Phase.
- 3.7 Overview of assembling process
- 3.8 Pass Structure of Assembler – One pass, Two pass assembler.
- 3.9 Problems of 1-pass assembler - forward reference, efficiency, Table of Incomplete Instructions.
- 3.10 Design of 2-pass Assembler – Pass-I and Pass-II
- 3.11 Data structure of 2-pass assembler.
- 3.12. Intermediate Code – Need, Forms-variant I and Variant II

**4. Macros and Macro Processors**

**[10]**

- 4.1 Definition
- 4.2 Macro definition and call
- 4.3 Macro expansion – positional and keyword parameters
- 4.4 Design of Data structures to be used for Macro definition and use
- 4.5 Nested macro calls
- 4.6 Advanced macro facilities – alteration of flow of control during expansion, expansion time variable, conditional expansion, expansion time loops. (with examples)
- 4.7 Design of macro preprocessor – Design overview, data structure, processing of macro definition and macro expansion (Except algorithms)

4.8 Macro assembler – Comparison of macro preprocessor and macro assembler. Pass structure of macro assembler.

**5. Compiler Design options** [2]

5.1 Interpreter - Use of interpreter, definition, Comparison with compiler, Overview of interpretation, Pure and impure interpreter.

5.2 P-code compiler

**6. Linker and Loader** [6]

6.1 Introduction

6.2 Concept of bindings, static and dynamic binding, translated, linked and load time addresses.

6.3 Relocation and linking concept – program relocation, performing relocation, public and external references, linking, binary program, object module.

6.4 Relocatability - nonrelocatable, relocatable, and self relocating programs (no algorithms), Linking for Overlays.

6.5 Object file formats: a.out, ELF, COFF, EXE, PE and COM

**7. Debuggers & Development utilities** [6]

7.1 Debugging functions and capabilities

7.2 Types of debuggers: visual & console -Case study of ddd(visual ) and gdb(console)

7.3 Development utilities on UNIX/Linux strip, make, nm, objdump, intermediate files in compilation process etc.

**8. Operating System as System Software** [6]

8.1 What Operating Systems Do – User View, System View, Defining OS

8.2 Computer System Architecture – Single processor system, Multiprocessor systems, Clustered Systems

8.3 Operating System Operations – Dual mode operation, Timer

8.4 Process Management

8.5 Memory Management

8.6 Storage Management – File system management, Mass storage management, Caching, I/O systems

8.7 Protection and Security

8.8 Distributed Systems

8.9 Special Purpose System – Real time embedded systems, Multimedia systems, Handheld systems,

8.10 Computer Environment – Traditional computing, Client server computing, Peer to peer Computing

**9. System Structure** [2]

9.1 Operating System Services

9.2 User Operating-System Interface – Command interpreter, GUI

9.3 System Calls

9.4 Types of System Calls – Process control, File management, Device management, Information maintenance, Communication, Protection

**Reference Books:**

1. Systems Programming and Operating Systems by D.M.Dhamdhare (Second Revised Edition). [Chapters: 2, 3, 4, 5, 7]

2. System Software - An introduction to Systems Programming - Leland L. Beck (Pearson Education) [ Chapter: 1]

3. Linkers and Loaders – John R. Levine, Elsevier Morgan Kaufmann[chapter 6]

4. Operating System Concepts - Siberchatz, Galvin, Gagne (8th Edition).[chapter 8, 9]

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Operating Systems**  
**Code No. : CS-341**

---

**Semester IV**

**Total Lectures : 48**

**Aim :** To understand the design and implementation issues of Operating System.

Objectives :

- To understand design issues related to process management and various related algorithms
- To understand design issues related to memory management and various related algorithms
- To understand design issues related to File management and various related algorithms

**1. Introduction** **[2]**

- 1.1 Operating System Structure – Simple structure, Layered approach, Micro kernels, Modules
- 1.2 Virtual Machines – Introduction, Benefits
- 1.3 System Boot

**2. Process Management** **[4]**

- 2.1 Process Concept – The process, Process states, Process control block.
- 2.2 Process Scheduling – Scheduling queues, Schedulers, context switch
- 2.3 Operations on Process – Process creation with program using fork(), Process termination
- 2.4 Interprocess Communication – Shared memory system, Message passing systems.

**3. Multithreaded Programming** **[2]**

- 3.1 Overview
- 3.2 Multithreading Models

**4. Process Scheduling** **[8]**

- 4.1 Basic Concept – CPU-I/O burst cycle, CPU scheduler, Preemptive scheduling, Dispatcher
- 4.2 Scheduling Criteria
- 4.3 Scheduling Algorithms – FCFS, SJF, Priority scheduling, Round-robin scheduling, Multiple queue scheduling, Multilevel feedback queue scheduling
- 4.4 Thread Scheduling

**5. Process Synchronization** **[6]**

- 5.1 Background
- 5.2 Critical Section Problem
- 5.3 Semaphores: Usage, Implementation
- 5.4 Classic Problems of Synchronization – The bounded buffer problem, The reader writer problem, The dining philosopher problem

**6. Deadlocks** **[8]**

- 6.1 System model
- 6.2 Deadlock Characterization – Necessary conditions, Resource allocation graph
- 6.3 Deadlock Prevention
- 6.4 Deadlock Avoidance - Safe state, Resource allocation graph algorithm, Banker's Algorithm
- 6.5 Deadlock Detection
- 6.6 Recovery from Deadlock – Process termination, Resource preemption

**7. Memory Management** **[11]**

- 7.1. Background – Basic hardware, Address binding, Logical versus physical address space, Dynamic loading, Dynamic linking and shared libraries

7.2 Swapping

7.3 Contiguous Memory Allocation – Memory mapping and protection, Memory allocation, Fragmentation

7.4 Paging – Basic Method, Hardware support, Protection, Shared Pages

7.5 Segmentation – Basic concept, Hardware

7.6 Virtual Memory Management – Background, Demand paging, Performance of demand paging, Page replacement – FIFO, OPT, LRU, Second chance page replacement

## **8. File System**

[7]

8.1 File concept

8.2 Access Methods – Sequential, Direct, Other access methods

8.3 Directory and Disk Structure – Storage structure, Directory overview, Single level directory, Two level directory, Tree structure directory, Acyclic graph directory, General graph directory

8.4 Allocation Methods – Contiguous allocation, Linked allocation, Indexed allocation

8.5 Free Space Management – Bit vector, Linked list, Grouping, Counting, Space maps

### **Reference Books:**

1. Operating System Concepts - Silberchatz, Galvin, Gagne (8th Edition).

2. Operating Systems : Principles and Design – Pabitra Pal Choudhary (PHI Learning Private Limited)

---

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Theoretical Computer Science**  
**Code No. : CS-332**

---

**Semester III**

**Total Lectures : 48**

**Aim:**

To have a introductory knowledge of automata, formal language theory and computability.

**Objectives:**

- To have an understanding of finite state and pushdown automata.
- To have a knowledge of regular languages and context free languages.
- To know the relation between regular language, context free language and corresponding recognizers.
- To study the Turing machine and classes of problems.

**Prerequisite:**

- Sets, Operations on sets, Finite & infinite sets Formal Language
- Relation, Equivalence Relation,(reflexive, transitive and symmetric closures)

**1. Introduction**

**[3]**

1.1 Symbol, Alphabet, String, Prefix & Suffix of Strings, Formal Language, Operations on Languages.

1.2 Regular Expressions (RE) : Definition & Example

1.3 Regular Expressions Identities.

**2. Finite Automata**

**[12]**

2.1 Deterministic finite Automaton – Definition, DFA as language recognizer, DFA as a pattern recognizer.

2.2 Nondeterministic finite automaton – Definition and Examples.

2.3 NFA TO DFA : Method (From Book 4)

2.4 NFA with  $\epsilon$ - transitions Definition and Examples.

2.5 NFA with  $\epsilon$ -Transitions to DFA & Examples

2.6 Finite automaton with output – Mealy and Moore machine, Definition and Examples.

2.7 Minimization of DFA, Algorithm & Problem using Table Method.

**3. Regular Languages**

**[5]**

3.1 Regular language-Definition and Examples.

3.2 Conversion of RE To FA-Examples.

3.3 Pumping lemma for regular languages and applications.

3.4 Closure properties of regular Languages

(Union, Concatenation, Complement, Intersection and Kleene closure)

**4. Context Free Grammar and Languages**

**[12]**

4.1 Grammar - Definition and Examples.

4.2 Derivation-Reduction - Definition and Examples.

4.3 Chomsky Hierarchy.

4.4 CFG : Definition & Examples. LMD, RMD, Parse Tree

4.5 Ambiguous Grammar : Concept & Examples.

4.6 Simplification of CFG :

4.6.1 Removing Useless Symbols,

4.6.2 Removing unit productions

4.6.3 Removing  $\epsilon$  productions & Nullable symbols

4.7 Normal Forms :

4.7.1 Chomsky Normal Form (CNF) Method & Problem

- 4.7.2 Greibach Normal form (GNF) Method & Problem
- 4.8 Regular Grammar : Definition.
  - 4.8.1 Left linear and Right Linear Grammar-Definition and Example.
  - 4.8.2 Equivalence of FA & Regular Grammar
    - 4.8.2.1 Construction of regular grammar equivalent to a given DFA
    - 4.8.2.2 Construction of a FA from the given right linear grammar
- 4.9 Closure Properties of CFL's(Union, concatenation and Kleen closure) Method and examples

## **5. Push Down Automaton**

[6]

- 5.1 Definition of PDA and examples
- 5.2 Construction of PDA using empty stack and final State method : Examples using stack method
- 5.3 Definition DPDA & NPDA, their correlation and Examples of NPDA
- 5.4 CFG (in GNF) to PDA : Method and examples

## **6. Turing Machine**

[10]

- 6.1 The Turing Machine Model and Definition of TM
- 6.2 Design of Turing Machines
- 6.3 Problems on language recognizers.
- 6.4 Language accepted by TM
- 6.5 Types of Turing Machines(Multitrack TM,Two way TM, Multitape TM,Non-deterministic TM)
- 6.6 Introduction to LBA (Basic Model) &CSG.( Without Problems)
- 6.7 Computing TM, Enumerating TM, Universal TM
- 6.8 Recursive Languages
  - 6.5.1. Recursive and Recursively enumerable Languages.
  - 6.5.2. Difference between recursive and recursively enumerable language.
- 6.9 Turing Machine Limitations
- 6.10 Decision Problem, Undecidable Problem, Halting Problem of TM

## **References :-**

- 1 Introduction to Automata theory, Languages and computation By John E. Hopcroft and Jeffrey Ullman – Narosa Publishing House.
2. Introduction to Automata theory, Languages and computation By John Hopcroft, Rajeev Motwani and Jeffrey Ullman –Third edition Pearson Education
3. Introduction to Computer Theory Daniel I. A. Cohen – 2<sup>nd</sup> edition – John Wiley & Sons
4. Theory of Computer Science (Automata, Language & Computation) K. L. P. Mishra & N. Chandrasekaran, PHI Second Edition
5. Introduction to Languages and The Theory of Computation John C. Martin TMH, Second Edition

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Compiler Construction**  
**Code No. : CS-342**

---

**Semester IV**

**Total Lectures : 48**

**Aim :**

To understand the various phases of a compiler and to develop skills in designing a compiler

**Objective :**

- To understand design issues of a lexical analyzer and use of Lex tool
- To understand design issues of a parser and use of Yacc tool
- To understand issues related to memory allocation
- To understand and design code generation schemes

**1. Introduction**

**[5]**

- 1.1 Definition of Compiler, Aspects of compilation.
- 1.2 The structure of Compiler.
- 1.3 Phases of Compiler – Lexical Analysis, Syntax Analysis, Semantic Analysis, Intermediate Code generation, code optimization, code generation.
- 1.4 Error Handling
- 1.5 Introduction to one pass & Multipass compilers, cross compiler, Bootstrapping.

**2. Lexical Analysis(Scanner)**

**[5]**

- 2.1 Review of Finite automata as a lexical analyzer,
- 2.2 Applications of Regular Expressions and Finite Automata ( lexical analyzer, searching using RE), Input buffering, Recognition of tokens
- 2.3 LEX: A Lexical analyzer generator (Simple Lex Program)

**3. Syntax Analysis(Parser)**

**[20]**

- 3.1 Definition , Types of Parsers
- 3.2 Top-Down Parser –
  - 3.2.1 Top-Down Parsing with Backtracking: Method & Problems
  - 3.2.2 Drawbacks of Top-Down parsing with backtracking,
  - 3.2.3 Elimination of Left Recursion(direct & indirect)
  - 3.2.4 Need for Left Factoring & examples
- 3.3 Recursive Descent Parsing : Definition
  - 3.3.1 Implementation of Recursive Descent Parser Using Recursive Procedures
- 3.4 Predictive [LL(1)]Parser(Definition, Model)
  - 3.4.1 Implementation of Predictive Parser[LL(1)]
  - 3.4.2 FIRST & FOLLOW
  - 3.4.3 Construction of LL(1) Parsing Table
  - 3.4.4 Parsing of a String using LL(1) Table
- 3.5 Bottom-Up Parsers
- 3.6 Operator Precedence Parser -Basic Concepts
  - 3.6.1 Operator Precedence Relations form Associativity & Precedence
  - 3.6.2 Operator Precedence Grammar
  - 3.6.3 Algorithm for LEADING & TRAILING(with ex.)
  - 3.6.4 Algorithm for Operator Precedence Parsing (with ex.)
  - 3.6.5 Precedence Functions
- 3.7 Shift Reduce Parser
  - 3.7.1 Reduction, Handle, Handle Pruning
  - 3.7.2 Stack Implementation of Shift Reduce Parser ( with examples)

- 3.8 LR Parser
  - 3.8.1 Model
  - 3.8.2 Types [SLR(1), Canonical LR, LALR] Method & examples.
- 3.9 YACC (from Book 3) –program sections, simple YACC program for expression evaluation

#### **4. Syntax Directed Definition [8]**

- 4.1 Syntax Directed Definitions(SDD)
  - 4.1.1 Inherited & Synthesized Attributes
  - 4.1.2 Evaluating an SDD at the nodes of a Parse Tree, Example
- 4.2 Evaluation Orders for SDD's
  - 4.2.1 Dependency Graph
  - 4.2.2 Ordering the Evaluation of Attributes
  - 4.2.3 S-Attributed Definition
  - 4.2.4 L-Attributed Definition
- 4.3 Application of SDT
  - 4.3.1 Construction of syntax trees,
  - 4.3.2 The Structure of a Type
- 4.4 Translation Schemes
  - 4.4.1 Definition, Postfix Translation Scheme

#### **5. Memory Allocation [2]**

- 5.1 Memory allocation – static and dynamic memory allocation,
- 5.2 Memory allocation in block structure languages, Array allocation and access.

#### **6. Code Generation and Optimization [8]**

- 6.1 Compilation of expression –
  - 6.1.1 Concepts of operand descriptors and register descriptors with example.
  - 6.1.2 Intermediate code for expressions – postfix notations,
  - 6.1.3 triples and quadruples, expression trees.
- 6.2 Code Optimization – Optimizing transformations – compile time evaluation, elimination of common sub expressions, dead code elimination, frequency reduction, strength reduction
- 6.3 Three address code
  - 6.3.1 DAG for Three address code
  - 6.3.2 The Value-number method for constructing DAG's.
- 6.4 Definition of basic block, Basic blocks And flow graphs
- 6.5 Directed acyclic graph (DAG) representation of basic block
- 6.6 Issues in design of code generator

#### **References :-**

1. Compilers: Principles, Techniques, and Tools ,Alfred V. Aho, Ravi Sethi, Jeffrey D. Ullman
2. Principles of Compiler Design By : Alfred V. Aho, Jeffrey D. Ullman (Narosa Publication House)
3. LEX & YACC (O'reilly Publication)

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Computer Networks -I**  
**Code No. : CS-333**

---

**Semester III**

**Total Lectures : 48**

**Pre-requisites:** Basics of computer, Knowledge of 'C' for assignment.

**Objectives:** This course will prepare students in

- Basic networking concepts.
- Understand different types of networks, various topologies and application of networks.
- Understand types of addresses, data communication.
- Understand the concept of networking models, protocols, functionality of each layer.
- Learn basic networking hardware and tools.

**1. Introduction to Computer Networks** **[5]**

1.1 Computer Networks- Goals and applications – Business Application , Home Application, Mobile User, Social Issues [1, TAN. Ch. 1]

1.2 Network Hardware - Broadcast and point-to-point, topologies – star, bus, mesh, ring etc. [1, TAN. Ch. 1]

1.3 Network Types-LAN, MAN, WAN, Wireless Networks, Home Networks, Internetworks, Protocols and Standards –Definition of Protocol, Defacto and Dejure standard, Peer –to-peer and Server – based LAN [3, TAN. Ch. 1]

1.4 Network Software Protocol Hierarchies -layers, protocols, peers, interfaces [2, TAN. Ch. 1]

1.5 Network architecture, protocol stack design issues of the layers –addressing, error control, flow control, multiplexing and demultiplexing, routing Connection-oriented and connectionless service ,Service Primitives – listen, connect, receive, send, disconnect and Berkley Socket ,he relationships of services to protocol.

**2. Network Models** **[5]**

2.1 OSI Reference, Model, Functionality of each layer [2 FORO. Ch2]

2.2 TCP/IP Reference, Model, Introduction to IP, TCP, and UDP, TCP/IP Protocol Suite[1 FORO. Ch2]

2.3 Comparison of OSI and TCP/IP model [1 FORO. Ch2]

2.4 Addressing Physical, Logical and Port addresses [1 FORO. Ch2]

**3. Transmission Media** **[4]**

3.1 Guided Media - Twisted pair cable, Coaxial cable, Fiber Optic cable[ FORO. Ch7]

3.2 Unguided Media – Wireless- Radio Waves,- Microwaves, Infrared [ FORO. Ch7], Lightwave transmission [TAN. Ch. 2]

3.3 Types of cabling and Networking Tool - CAT5 and CAT6 Cable Color Code, Crossover Cabling and Straight Through Cable, Crimping and Line testing tool

**4. The Physical Layer** **[13]**

4.1 Basic Concepts Signals, Types – Analog and Digital Signals, Bit rate, bit length, baseband transmission

4.2 Transmission Impairments –attenuation, distortion and noise, Data Rate Limits – Nyquist's bit rate formula for noiseless channel, and Shannon's law (**Enough problems should be covered on every topic.**) [3 FORO. Ch. 3 ]

- 4.3 Performance of the Network Bandwidth, Throughput, Latency(Delay), Bandwidth –Delay, Product, Jitter (**Enough problems should be covered on every topic.**)[1 FORO. Ch3]
- 4.4 Line Coding Characteristics, Line Coding, Schemes – Unipolar, NRZ, RZ, Manchester and Differential Manchester [2 FORO Ch.4]
- 4.5 Transmission Modes, Parallel Transmission Serial Transmission –Asynchronous and Synchronous[1 FORO. Ch. 4]
- 4.6 Multiplexing FDM and TDM, Switching Circuit Switching, Message Switching and Packet Switching[1 TAN. Ch. 2]
- 4.7 Physical Layer Devices- Physical Layer Devices Repeaters, Hubs [FORO. Ch.15]
- 4.8 ISDN Services, Evolution, Architecture [2 TAN. Ch. 2]

## **5. The Data Link Layer**

[4]

- 5.1 Design Issues Services to Network Layer, Flow Control, Error Control 1Framing Character Count, Byte Stuffing, Bit Stuffing and Physical Layer Coding Violations[2 TAN Ch. 3]  
Error Control Hamming Code and CRC(**Enough problems should be covered on every topic.**) [2 TAN Ch . 3.]
- 5.2 Elementary Data Link Layer Protocols Utopia, A Simplex Stop-And-Wait, A Simplex protocol for noisy channel [2 TAN Ch. 3]
- 5.3 Sliding Window Protocols Piggybacking-Need, Advantages/Disadvantages, 1-bit sliding window protocols, Pipelining –Go-Back N and Selective Repeat [3 TAN Ch. 3]
- 5.4 Data Link Layer Protocols-HDLC – frame format, all frame types PPP – Use, Frame Format, Use of PPP in the Internet [2 TAN Ch. 3]
- 5.5 Data Link Layer Devices Bridges – Transparent and Source Routing Bridges, Bridges Connecting Different LANs [2 FORO. Ch.15]

## **6 The Medium Access Sublayer**

[7]

- 6.1 Random Access Protocols
- 6.2 ALOHA – pure and slotted
- 6.3 CSMA – 1-persistent, p-persistent and non-persistent CSMA/CD, CSMA/CA3 [FORO. Ch. 12]
- 6.4 Controlled Access Reservation, Polling and Token Passing [1 FORO. Ch. 12]
- 6.5 Channelization FDMA, TDMA and CDMA-Analogy, Idea, Chips, Data Representation, Encoding and Decoding, Signal Level, Sequence Generation(**Enough problems should be covered on every topic.**)[3 FORO. Ch.12]

### **Reference Books:**

- 1) Computer Networks by Andrew Tanenbaum, Pearson Education.
- 2) Data Communication and Networking by Behrouz Forouzan, TATA McGraw Hill.
- 3) Computer Networks by Natalia and Victor Olifer, Wiley Publications.
- 4) TCP/IP Protocol Suite-Third Edition, Behrouz Forouzan, TATA McGraw Hill.

### **Guidelines For Examination:**

- 1) Frame and Packet formats should be asked.
- 2) Problems should be asked at least for 8 marks.

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Computer Networks -II**  
**Code No. : CS-343**

---

**Semester IV**

**Total Lectures : 48**

**Pre-requisites:** Basics of computer networks covered last semester, Knowledge of 'C'.

**Objectives:** This course will prepare students in

- Basic networking concepts.
- Understand wired and wireless networks ,its types, functionality of layer.
- Understand importance of network security and cryptography.

**1. Wired LANS**

**[10]**

- 1.1 IEEE Standards Data Link Layer, Physical Layer 1
- 1.2 Standard Ethernet MAC Sublayer – Frame Format, Frame Length, Addressing, Access Method
- 1.4 Physical Layer – Encoding and Decoding, 10Base5, 10Base2, 10Base-T, 10Base-F, Changes In The Standard – Bridged Ethernet, Switched Ethernet, Full Duplex Ethernet Fast Ethernet – Goals, MAC Sublayer,Topology, Implementation Gigabit Ethernet – goals, MAC Sublayer, Topology, Implementation Ten-Gigabit Ethernet – goals, MAC Sublayer, Physical Layer [6 FORO. Ch.13]
- 1.5 Backbone Networks Bus Backbone, Star Backbone, Connecting Remote LANs [1 FORO. Ch.15]
- 1.6 Virtual LANs Membership, Configuration, Communication between Switches, Advantages

**2. Wireless LAN**

**[1]**

- 2.1 IEEE 802.11 Architecture – Basic Service Set, Extended Service Set[1 FORO.Ch.14]
- 2.2 Bluetooth Architecture – Piconet, scatternet [1 FORO.Ch. 14]

**3. The Network Layer**

**[10]**

- 3.1 Design Issues Store-and-forward packet switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection Oriented Service, Comparison of Virtual Circuit and Datagram [3 TAN.Ch.5]
- 3.2 Logical Addressing IPV4 Addresses – *Address Space, Notations, Classful Addressing, Subnetting, Supernetting, Classless Addressing, Network Address Translation(NAT)* [3 FORO.Ch. 19](**Enough problems should be covered on Addressing**), IPV4 Protocol Datagram Format, Fragmentation, Checksum, Options [3 FORO.Ch. 20], Routing Properties of routing algorithm, Comparison of Adaptive and Non-Adaptive Routing Algorithms [1 TAN.Ch. 5]
- 3.3 Congestion Control General Principles of Congestion Control, Congestion Prevention Policies [**2 FORO. Ch. 24 page 765**]
- 3.4 Network Layer Devices -Routers [1 FORO. Ch.15]

**4. Address Mapping**

**[4]**

- 4.1 Protocol(ARP)-Cache Memory, Packet Format, Encapsulation, Operation, Four Different Cases, Proxy ARP, RARP , BOOTP, DHCP – Static Address Allocation, Dynamic Address Allocation, Manual and automatic Configuration [4 FORO. Ch.21]

**5. The Transport Layer**

**[6]**

- 5.1 Process-to-Process Delivery Client Server Paradigm, Multiplexing and Demultiplexing, Connectionless Vs Connection-Oriented Service, Reliable Vs Unreliable [2 FORO.Ch.23]
- 5.2 User Datagram Protocol(UDP) Datagram Format, Checksum, UDP operations, Use of UDP[2 FORO.Ch.23]

5.3 Transmission Control Protocol (TCP) TCP Services – Process to-Process Communication, Stream Delivery Service, Full –Duplex Communication

5.4 TCP Features –Numbering System, Flow Control, Error Control, Congestion Control [2 FORO.Ch.2325]

5.5 TCP Segment – Format

## **6. The Application Layer**

[7]

6.1 Domain Name System (DNS) Name Space, Domain, Name Space, Distribution of Name Space, DNS in the Internet, Name –Address Resolution [2 FORO.Ch.25]

6.2 E-MAIL Architecture, User Agent, Message Transfer Agent-SMTP, Message Access Agent-POP,IMAP, Web Based Mail[2 FORO.Ch.26]

6.3 File Transfer Protocol (FTP) Communication over control connection, Communication over Data Connection, Anonymous FTP [1 FORO. Ch.26]

6.4 WWW Architecture, WEB Documents[1 FORO. Ch.27]

6.5 HTTP - HTTP Transaction, Persistent and Non persistent Connection, Proxy Server[1 FORO. Ch.27]

6.6 Devices- Gateways [FORO. Ch.15]

## **7. Network Security**

[10]

7.1 Introduction – Security Services- Message-Confidentiality, Integrity, Authentication, Non repudiation. Entity(User)- Authentication. [Foro. Ch 31-31.1]

7.2 Message confidentiality –Confidentiality with Asymmetric-Key Cryptography, Confidentiality with Symmetric-Key Cryptography [Foro. Ch 31-31.2]

7.3 Cryptography Encryption Model, Substitution Cipher and Transposition Cipher, Two Fundamental Cryptographic Principles (Problems should be covered.)[3 TAN. Ch. 8]

7.4 Communication Security Firewalls [1 TAN. Ch. 8]

7.5 Web Security Threats, Secure Naming,DNS Spoofing – DNS Spoofing, Secure DNS,Self Certifying names[2 TAN. Ch. 8]

7.6 Mobile Mode Security Java Applet Security, Activex, JavaScript, Viruses [2 TAN. Ch. 8] Social Issues Privacy, Anonymous Remailers, Freedom of Speech, Steganography [2 TAN. Ch. 8]

### **Reference Books:**

- 1) Computer Networks by Andrew Tanenbaum, Pearson Education.
- 2) Data Communication and Networking by Behrouz Forouzan, TATA McGraw Hill.
- 3) Computer Networks by Natalia and Victor Olifer, Wiley Publications.
- 4) TCP/IP Protocol Suite-Third Edition, Behrouz Forouzan, TATA McGraw Hill.

### **Guidelines For Examination:**

- 1) Frame and Packet formats should be asked.
- 2) Problems should be asked at least for 8 marks.

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Internet Programming I**  
**Code No. : CS-334**

---

**Semester III**

**Total Lectures: 48**

**Aim:** To Design dynamic and interactive Web pages.

**Objective:**

- Learn Core-PHP, Server Side Scripting Language
- Learn PHP-Database handling.

**Prerequisite:** HTML.

**1. Introduction to web techniques** **[8]**

- 1.1 HTTP basics, Introduction to Web server and Web browser
  - 1.2 Introduction to PHP
  - 1.3 What does PHP do?
  - 1.4 Lexical structure
  - 1.5 Language basics
- Book 1 chapter 2

**2. Function and String** **[10]**

- 2.1 Defining and calling a function
  - 2.2 Default parameters
  - 2.3 Variable parameters, Missing parameters
  - 2.4 Variable function, Anonymous function
  - 2.5 Types of strings in PHP
  - 2.6 Printing functions
  - 2.7 Encoding and escaping
  - 2.8 Comparing strings
  - 2.9 Manipulating and searching strings
  - 2.10 Regular expressions
- Book 1 chapter 3 and 4

**3. Arrays** **[6]**

- 3.1 Indexed Vs Associative arrays
  - 3.2 Identifying elements of an array
  - 3.3 Storing data in arrays
  - 3.4 Multidimensional arrays
  - 3.4 Extracting multiple values
  - 3.5 Converting between arrays and variables
  - 3.6 Traversing arrays
  - 3.7 Sorting
  - 3.8 Action on entire arrays
  - 3.9 Using arrays
- Book 1 chapter 5

**4. Introduction to Object Oriented Programming** **[8]**

- 4.1 Classes
- 4.2 Objects
- 4.3 Introspection
- 4.4 Serialization
- 4.5 Inheritance

4.6 Interfaces  
4.7 Encapsulation  
Book 1 , 2 chapter 12

## 5. Files and directories

[6]

5.1 Working with files and directories  
5.2 Opening and Closing, Getting information about file, Read/write to file,  
Splitting name and path from file, Rename and delete files  
5.3 Reading and writing characters in file  
5.4 Reading entire file  
5.5 Random access to file data  
5.6 Getting information on file  
5.7 Ownership and permissions  
Book 2 chapter 7

## 6. Databases (PHP-PostgreSQL)

[10]

6.1 Using PHP to access a database  
6.2 Relational databases and SQL  
6.3 PEAR DB basics  
6.4 Advanced database techniques  
6.5 Sample application (Mini project)  
Book 1 chapter 9

## References

1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe, O'Reilly publication
2. Beginning PHP 5 , Wrox publication
3. PHP web services, Wrox publication
4. AJAX Black Book, Kogent solution
5. Mastering PHP , BPB Publication
6. PHP cookbook, O'Reilly publication
7. PHP for Beginners, SPD publication
8. Programming the World Wide Web , Robert W Sebesta(3<sup>rd</sup> Edition)
9. Check out Joomla!press  
*Pearson (Addison-Wesley Professional).*
10. [www.php.net.in](http://www.php.net.in)
11. [www.W3schools.com](http://www.W3schools.com)
12. [www.wrox.com](http://www.wrox.com)
13. <https://api.drupal.org>

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Internet Programming II**  
**Code No. : CS-344**

---

**Semester IV**

**Total Lectures: 48**

**Aim:** To Design dynamic and interactive Web pages.

**Objective:**

- Learn different technologies used at client Side Scripting Language
- Learn XML,CSS and XML parsers.
- One PHP framework for effective design of web application.
- Learn JavaScript to program the behavior of web pages.
- Learn AJAX to make our application more dynamic.

**1. Web Techniques** **[10]**

- 1.1 Variables
  - 1.2 Server information
  - 1.3 Processing forms
  - 1.4 Setting response headers
  - 1.5 Maintaining state
  - 1.6 SSL
- Book 1 chapter 7

**2. Handling email with php** **[8]**

- 2.1 Email background
  - 2.2 Internet mail protocol
  - 2.3 Structure of an email message
  - 2.4 Sending email with php
  - 2.5 Email attachments.
  - 2.6 Email id validation and verification
  - 2.7 PHP error handling.
- Book 2 chapter 15

**3. PHP framework** **[4]**

- 3.1 Introduction to PHP framework.
  - 3.2 Features, Applications.
  - 3.3 One example like JOOMLA,DRUPAL.
- Book 11, <https://api.drupal.org>

**4. XML** **[8]**

- 4.1 What is XML?
  - 4.2 XML document Structure
  - 4.3 PHP and XML
  - 4.4 XML parser
  - 4.5 The document object model
  - 4.6 The simple XML extension
  - 4.7 Changing a value with simple XML
- Book 2 chapter 8

**5. WEB DESIGNING TECHNOLOGIES(JavaScript-DHTML)** **[10]**

- 5.1 Overview of JavaScript, DHTML
- 5.2 Object Orientation and JavaScript

- 5.3 Basic Syntax(JS datatypes, JS variables )
- 5.4 Primitives, Operations and Expressions
- 5.5 Screen Output and keyboard input(Verification and Validation)
- 5.6 JS Control statements
- 5.7 JS Functions
- 5.8 JavaScript HTML DOM Events(onmouseup, onmousedown, onclick, onload,onmouseover,onmouseout).
- 5.9 JS Strings.
- 5.10 JS String methods
- 5.11 JS popup boxes(alert, confirm, prompt).
- 5.12 Changing property value of different tags using DHTML  
(ex. adding innerhtml for DIV tag, changing source of image etc.).

Book 10, [www.w3schools.com](http://www.w3schools.com).

## 6. AJAX

[8]

- 6.1 Introduction of AJAX
  - 6.2 AJAX web application model
  - 6.3 AJAX –PHP framework
  - 6.4 Performing AJAX validation
  - 6.5 Handling XML data using php and AJAX
  - 6.6 Connecting database using php and AJAX
- Book 4 chapter 1,2 and 9

## References

1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe O'Reilly publication
2. Beginning PHP 5, Wrox publication
3. PHP web services , Wrox publication
4. AJAX Black Book Kogent solution
5. Mastering PHP BPB Publication
6. PHP cookbook O'Reilly publication
7. Learning PHP and MYSQL, O'Reilly publication
8. PHP and MYSQL, O'Reilly publication
9. PHP for Beginners, SPD publication
10. Programming the World Wide Web , Robert W Sebesta(3<sup>rd</sup> Edition)
11. Check out Joomla!presss **Pearson** (*Addison-Wesley Professional*).
12. [www.php.net.in](http://www.php.net.in)
13. [www.W3schools.com](http://www.W3schools.com)
14. [www.wrox.com](http://www.wrox.com)
15. <https://api.drupal.org>

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B.Sc. COMPUTER SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Programming in Java-I**  
**Code No. : CS-335**

---

**Semester IV**

**Total Lectures: 48**

**Aim:** To Design dynamic and interactive Web pages.

**Objective:**

- Learn different technologies used at client Side Scripting Language
- Learn XML,CSS and XML parsers.
- One PHP framework for effective design of web application.
- Learn JavaScript to program the behavior of web pages.
- Learn AJAX to make our application more dynamic.

**1. Web Techniques** **[10]**

- 1.1 Variables
  - 1.2 Server information
  - 1.3 Processing forms
  - 1.4 Setting response headers
  - 1.5 Maintaining state
  - 1.6 SSL
- Book 1 chapter 7

**2. Handling email with php** **[8]**

- 2.1 Email background
  - 2.2 Internet mail protocol
  - 2.3 Structure of an email message
  - 2.4 Sending email with php
  - 2.5 Email attachments.
  - 2.6 Email id validation and verification
  - 2.7 PHP error handling.
- Book 2 chapter 15

**3. PHP framework** **[4]**

- 3.1 Introduction to PHP framework.
  - 3.2 Features, Applications.
  - 3.3 One example like JOOMLA,DRUPAL.
- Book 11, <https://api.drupal.org>

**4. XML** **[8]**

- 4.1 What is XML?
  - 4.2 XML document Structure
  - 4.3 PHP and XML
  - 4.4 XML parser
  - 4.5 The document object model
  - 4.6 The simple XML extension
  - 4.7 Changing a value with simple XML
- Book 2 chapter 8

**5. WEB DESIGNING TECHNOLOGIES(JavaScript-DHTML)** **[10]**

- 5.1 Overview of JavaScript, DHTML

- 5.2 Object Orientation and JavaScript
- 5.3 Basic Syntax(JS datatypes, JS variables )
- 5.4 Primitives, Operations and Expressions
- 5.5 Screen Output and keyboard input(Verification and Validation)
- 5.6 JS Control statements
- 5.7 JS Functions
- 5.8 JavaScript HTML DOM Events(onmouseup, onmousedown, onclick, onload,onmouseover,onmouseout).
- 5.9 JS Strings.
- 5.10 JS String methods
- 5.11 JS popup boxes(alert, confirm, prompt).
- 5.12 Changing property value of different tags using DHTML  
(ex. adding innerhtml for DIV tag, changing source of image etc.).

Book 10, [www.w3schools.com](http://www.w3schools.com).

## 6. AJAX

[8]

- 6.1 Introduction of AJAX
  - 6.2 AJAX web application model
  - 6.3 AJAX –PHP framework
  - 6.4 Performing AJAX validation
  - 6.5 Handling XML data using php and AJAX
  - 6.6 Connecting database using php and AJAX
- Book 4 chapter 1,2 and 9

## References

1. Programming PHP By Rasmus Lerdorf and Kevin Tatroe O'Reilly publication
2. Beginning PHP 5, Wrox publication
3. PHP web services , Wrox publication
4. AJAX Black Book Kogent solution
5. Mastering PHP BPB Publication
6. PHP cookbook O'Reilly publication
7. Learning PHP and MYSQL, O'Reilly publication
8. PHP and MYSQL, O'Reilly publication
9. PHP for Beginners, SPD publication
10. Programming the World Wide Web , Robert W Sebesta(3<sup>rd</sup> Edition)
11. Check out Joomla!press     **Pearson** (*Addison-Wesley Professional*).
12. [www.php.net.in](http://www.php.net.in)
13. [www.W3schools.com](http://www.W3schools.com)
14. [www.wrox.com](http://www.wrox.com)
15. <https://api.drupal.org>

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B.Sc. COMPUTER SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Programming in Java-II**  
**Code No. : CS-345**

---

---

**Semester IV**

**Total Lectures : 48**

**Prerequisite:**

- Knowledge of Core Java (CS – 345)

**Objectives:**

- To learn database programming using Java
- To study web development concept using Servlet and JSP
- To develop a game application using multithreading
- To learn socket programming concept

**1. Collection**

**[6]**

- 1.1 Introduction to the Collection framework
- 1.2 List – ArrayList, LinkedList and Vector, Stack, Queue
- 1.3 Set - HashSet, TreeSet, and LinkedHashSet
- 1.4 Map – HashMap, LinkedHashMap, Hashtable and TreeMap
- 1.5 Interfaces such as Comparator, Iterator, ListIterator, Enumeration

**2. Database Programming**

**[10]**

- 2.1 The design of jdbc, jdbc configuration
- 2.2 Types of drivers
- 2.3 Executing sql statements, query execution
- 2.4 Scrollable and updatable result sets
- 2.5 Metadata – DatabaseMetadata, ResultSetMetadata
- 2.6 Transactions – commit(), rollback(), SavePoint  
(Database : PostgreSQL)

**3. Servlet**

**[12]**

- 3.1 Introduction to Servlet and Hierarchy of Servlet
- 3.2 Life cycle of servlet
- 3.3 Tomcat configuration (Note: Only for Lab Demonstration)
- 3.4 Handling get and post request (HTTP)
- 3.5 Handling a data from HTML to servlet
- 3.6 Retriving a data from database to servlet
- 3.7 Session tracking – User Authorization, URL rewriting, Hidden form fields, Cookies and HttpSession

**4. JSP**

**[10]**

- 4.1 Simple first JSP program
- 4.2 Life cycle of JSP
- 4.2 Implicit Objects
- 4.3 Scripting elements – Declarations, Expressions, Scriptlets, Comments
- 4.4 JSP Directives – Page Directive, include directive
- 4.5 Mixing Scriptlets and HTML
- 4.6 Example of forwarding contents from database to servlet, servlet to JSP and displaying it using JSP scriptlet tag

## **5. Multithreading**

[6]

- 5.1 What are threads?
- 5.2 Life cycle of thread
- 5.3 Running and starting thread using Thread class
- 5.4 Thread priorities
- 5.5 Running multiple threads
- 5.6 The Runnable interface
- 5.7 Synchronization and interthread communication

## **6. Networking**

[4]

- 6.1 Networking basics – Protocol, Addressing, DNS, URL, Socket, Port
- 6.2 The java.net package – InetAddress, URL, URLConnection class
- 6.3 SocketServer and Socket class
- 6.4 Creating a Socket to a remote host on a port (creating TCP client and server)
- 6.5 Simple Socket Program Example

### **References:**

- 1) Complete reference Java by Herbert Schildt(5th edition)
- 2) Java 2 programming black books, Steven Horlzner
- 3) Programming with Java , A primer ,Forth edition , By E. Balagurusamy
- 4) Core Java Volume-I-Fundamentals, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press
- 5) Core Java Volume-II-Advanced Features, Eighth Edition, Cay S. Horstmann, Gary Cornell, Prentice Hall, Sun Microsystems Press

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : Object Oriented Software Engineering**  
**Code No. : CS-336**

---

**Semester III**

**Total Lectures: 48**

**Prerequisites**

- Knowledge of Object Oriented Concepts
- Knowledge of Classical Software Engineering

**Aim**

To Understand Object Oriented Modeling techniques and their applicability.

**Objectives**

- Understanding importance of Object Orientation in Software engineering
- Understand the components of Unified Modeling Language
- Understand techniques and diagrams related to structural modeling
- Understand techniques and diagrams related to behavioral modeling
- Understand techniques of Object Oriented analysis, design and testing

**1. Object Oriented Concepts and Principles**

**[4]**

- 1.1 What is Object Orientation ? - Introduction , Object , Classes and Instance , Polymorphism, Inheritance
- 1.2 Object Oriented System Development- Introduction, Function/Data Methods (With Visibility), Object Oriented Analysis, Object Oriented Construction
- 1.3 Identifying the Elements of an Object Model
- 1.4 Identifying Classes and Objects
- 1.5 Specifying the Attributes (With Visibility)
- 1.6 Defining Operations
- 1.7 Finalizing the Object Definition

**2. Introduction to UML**

**[2]**

- 2.1 Concept of UML
- 2.2 Advantages of UML

**3. Basic Structural Modeling**

**[5]**

- 3.1 Classes
- 3.2 Relationship
- 3.3 Common Mechanism
- 3.4 Class Diagram (Minimum three examples should be covered)

**4. Advanced Structural Modeling**

**[7]**

- 4.1 Advanced Classes
- 4.2 Advanced Relationship
- 4.3 Interface
- 4.4 Types and Roles
- 4.5 Packages
- 4.6 Object Diagram (Minimum three examples should be covered)

**5. Basic Behavioral Modeling**

**[9]**

- 5.1 Interactions
- 5.2 Use Cases and Use Case Diagram with stereo types (Minimum three examples should be covered)
- 5.3 Interaction Diagram (Minimum two examples should be covered)
- 5.4 Sequence Diagram (Minimum two examples should be covered)
- 5.5 Activity Diagram (Minimum two examples should be covered)
- 5.6 State Chart Diagram (Minimum two examples should be covered)

## **6. Object Oriented Analysis**

[6]

- 6.1 Iterative Development and the Rational Unified Process
- 6.2 Inception
- 6.3 Understanding Requirements
- 6.4 Use Case Model From Inception to Elaboration
- 6.5 Elaboration

## **7. Object Oriented Design**

[4]

- 7.1 The Booch Method, The Coad and Yourdon Method and Jacobson Method and Raumbaugh Method
- 7.2 The Generic Components of the OO Design Model
- 7.3 The System Design Process - Partitioning the Analysis Model, Concurrency and Sub System Allocation, Task Management Component, The Data Management Component, The Resource Management Component, Inter Sub System Communication
- 7.4 Object Design Process

## **8. Architectural modeling**

[6]

- 8.1 Component
- 8.2 Components Diagram (Minimum two examples should be covered)
- 8.3 Deployment Diagram (Minimum two examples should be covered)
- 8.4 Collaboration Diagram (Minimum two examples should be covered)

## **9. Object Oriented Testing**

[5]

- 9.1 Object Oriented Testing Strategies
- 9.2 Test Case Design for Object Oriented Software
- 9.3 Inter Class Test Case Design  
(Use of any freeware designing tool)

## **References.**

1. Grady Booch, James Rumbaugh, The Unified Modeling Language User/Reference Guide, Pearson Education INC
2. Ivar Jacobson, Object Oriented Software Engineering, Pearson Education INC
3. Craig Larman, Applying UML and Patterns Pearson Education INC
4. Bennett, Simon, Object Oriented Analysis and Design McGraw Hill

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER :Computer Graphics**  
**Code No. : CS-346**

---

**Semester IV**

**Total Lectures: 48**

**Pre – Requisites**

- Computer programming skills in C programming language
- Basic understanding of use of data structures
- Basic Mathematical concepts related to matrices and geometry

**Objectives**

- To study how graphics objects are represented in Computer
- To study how graphics system in a computer supports presentation of graphics information
- To study how interaction is handled in a graphics system
- To study how to manipulate graphics object by applying different transformations
- To provide the programmer's perspective of working of computer graphics

**1. Introduction to Computer graphics** **[4]**

- 1.1 Introduction to computer graphics & graphics systems
- 1.2 Components of Computer Graphics Representation, Presentation , Interaction and Transformations
- 1.3 Applications of Computer Graphics
- 1.3 Pixel/Point ,Raster v/s Vector ,RGB color model, intensity
- 1.4 Programming essentials – event driven programming. OpenGL library

**2. Input devices and Interaction tasks** **[4]**

- 2.1 Logical Interaction – Locator, valuator , pick and choice;
- 2.2 Physical devices used for interaction – keyboard, mouse, trackball,spaceball, tablets, light pen, joy stick, touch panel, data glove;
- 2.4 Keyboard , Mouse interaction in OpenGL
- 2.5 Graphical User Interfaces- cursors , radio buttons, scroll bars, menus, icons
- 2.6 Implementing GUI in open GL

**3. Presentation and Output devices** **[4]**

- 3.1 Presentation Graphics - frame buffer, display file, lookup table;
- 3.2 Display devices, Random and Raster scan display devices; CRT,
- 3.3 Hardcopy devices - Plotters and Printers

**4. Raster Scan Graphics** **[10]**

- 4.1 Line drawing algorithms; DDA algorithm, Bresenham's line drawing algorithm, Circle generation algorithm;
- 4.2 Scan conversions- Generation of the Display, Image compression
- 4.3 Displaying Lines and characters
- 4.3 Polygon filling -Scan converting polygons, fill algorithms, Boundary fill algorithm, flood fill algorithm

**5. Transformations** **[7]**

- 5.1 Basic transformations: translation, rotation, scaling; Matrix representations & homogeneous coordinates, Reflection, shear
- 5.2 Transformation of points, lines, parallel lines, intersecting lines. Viewing pipeline
- 5.3 Window to viewport co-ordinate transformation. Setting window and viewport in OpenGL.

## **6 Clipping** [7]

- 6.1 clipping operations , point clipping ,
- 6.2 Line clipping; Cohen Sutherland algorithm, Midpoint subdivision algorithm, Cyrus beck algorithm;
- 6.3 Polygon clipping , Sutherland Hodgman algorithm, Weiler-Atherton Algorithm

## **7 3D transformation & viewing** [6]

- 7.1 3D transformations: translation, rotation, scaling & other transformations;
- 7.2 Three dimensional viewing, Parallel and Perspective projections,
- 7.3 View Volumes and General Projection Transformations.
- 7.4 3 D clipping

## **8 Hidden surfaces Elimination** [4]

- 8.1 Depth comparison, A-buffer algorithm, Back face detection; Depth -Buffer
- 8.2 Scan-line Method - BSP tree method, the Painter's algorithm, Area-subdivision algorithm;

### **Text Books:**

1. Hearn, Baker – “ Computer Graphics ( C version 2nd Ed.)” – Pearson education
2. F. S. Hill, Stephen Kelly, Computer Graphics using OpenGL, PHI Learning
3. David F. Rogers - Procedural Elements of Computer Graphics, Tata McGRAW Hill

### **Reference Books:**

4. Foley, Vandam, Feiner, Hughes – “Computer Graphics principles (2nd Ed.) – Pearson Education.
5. W. M. Newman, R. F. Sproull – “Principles of Interactive computer Graphics” – TMH.
6. D. F. Rogers, J. A. Adams – “ Mathematical Elements for Computer Graphics (2nd Ed.)” – TMH
7. Z. Xiang, R. Plastock – “ Schaum's outlines Computer Graphics (2nd Ed.)” – TMH

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER : System Programming & Operating System**  
**Code No. : CS-347**

---

---

**Aim:**

To understand the process of designing and implementing System programs and operating system components.

**Objective :-**

1. Design and implement System programs with minimal features to understand their complexity.
2. Design and implement simulations of operating system level procedures.

**Syllabus**

Sr. No	Topic	Lectures
1	Line Editor	8 lectures
2	SMAC0 simulator	8 lectures
3	Assembler	12 Lectures
4	Macro processor	12 lectures
5	DFA driver	8 lectures
6	Development Utilities	8 lectures
7	Toy shell	8 Lectures
8	CPU Scheduler	12 lectures
9	Deadlock detection	8 lectures
10	Page Replacement Algorithms	12 lectures
11	File Allocation methods	12 Lectures

**Examination**

Internal Marks : Activity + Labbook(10+10)

External Marks : two programs(35each) oral(5) Activity(5)

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER :Lab Course II – Programming in Java**  
**Code No. : CS-348**

---

---

**Aim:**

To understand the process of designing and implementing Core and Advanced Java programs.

**Objective :-**

1. Implement core Java programs to solve simple problems
2. Implement Client and Server end Java programs

**Syllabus**

Sr. No	Topic	Lectures
<b>Core and Advanced Java</b>		
1	Simple Java programs	8 Lectures
2	Arrays and Packages	8 Lectures
3	Inheritance and Interfaces	8 Lectures
4	Exception Handling	8 Lectures
5	File Handling	8 Lectures
6	GUI designing & Event Handling	8 Lectures
7	Database Programming	8 Lectures
8	Multithreading	4 Lectures
9	Collection	8 Lectures
10	Servlets	8 Lectures
11	JSP	8 Lectures
12	Socket Programming	4 Lectures
<b>Computer Graphics</b>		
1	Simple Graphics program using OpenGL	4 Lectures
2	Using graphics primitives to display graphics	4 Lectures
3	Window to viewport transformations and other transformations	4 Lectures
4	Using simple Keyboard and Mouse interaction	4 Lectures
5	Graphics Mini project	16 Lectures

**Examination**

Internal Marks : Activity(CG) + Seminar(Enhanced java+ listening) (10+10)

External Marks : two programs(30each) oral(5) Activity(5)+ Labbook(10)

**SAVITRIBAI PHULE PUNE UNIVERSITY**  
**T.Y. B. Sc. COMPUTER SCIENCE SYLLABUS**  
**TO BE IMPLEMENTED FROM ACADEMIC YEAR 2015-16**  
**TITLE OF PAPER :Lab Course III – Programming in PHP & Project**  
**Code No. : CS-349**

---

---

**Aim:**

To understand the process of designing and implementing Core and Advanced Java programs.

**Objective :-**

1. Implement core Java programs to solve simple problems
2. Implement Client and Server end Java programs

**Syllabus**

Sr. No	Topic	Lectures
<b>Core and Advanced Java</b>		
1	Simple Java programs	8 Lectures
2	Arrays and Packages	8 Lectures
3	Inheritance and Interfaces	8 Lectures
4	Exception Handling	8 Lectures
5	File Handling	8 Lectures
6	GUI designing & Event Handling	8 Lectures
7	Database Programming	8 Lectures
8	Multithreading	4 Lectures
9	Collection	8 Lectures
10	Servlets	8 Lectures
11	JSP	8 Lectures
12	Socket Programming	4 Lectures
<b>Computer Graphics</b>		
1	Simple Graphics program using OpenGL	4 Lectures
2	Using graphics primitives to display graphics	4 Lectures
3	Window to viewport transformations and other transformations	4 Lectures
4	Using simple Keyboard and Mouse interaction	4 Lectures
5	Graphics Mini project	16 Lectures

**Examination**

Internal Marks : Activity(CG) + Seminar(Enhanced java+ listening) (10+10)

External Marks : two programs(30each) oral(5) Activity(5)+ Labbook(10)