

UNIVERSITY OF MUMBAI



Syllabus for the V and VI Semester
Program: B.Sc.
Course: Computer Programming and
System Analysis
(APPLIED COMPONENT)

(Credit Based Semester and Grading System with
effect from the academic year 2013–2014)

Syllabus for V and VI Semester APPLIED COMPONENT
Computer Programming And System Analysis
Based on Credit Based and Grading System

| SEMESTER V | | |
|----------------|--|-------------|
| Course code | Title | Credits |
| USACCS501 | Computer Programming and System Analysis -I | Two |
| Unit I | <p>Introduction to C Programming</p> <p>(a) Structure of C program: Header and body, Concept of header files, Use of comments, Compilation of a program.</p> <p>(b) Data Concepts: Variables, Constants, data types like: int, float char, double and void. Qualifiers: short and long size qualifiers, signed and unsigned qualifiers. Declaring variables, Scope of the variables according to block, Hierarchy of data types.</p> <p>(c) Types of operators: Arithmetic, Relational, Logical, Compound Assignment, Increment and decrement, Conditional or ternary operators. Precedence and order of evaluation. Statements and Expressions.</p> <p>(d) Mathematical functions : sin(), cos(), tan(), exp(), ceil(), floor(), log(), log10(), pow(), sqrt().</p> <p>(e) Type conversions: Automatic and Explicit type conversion.</p> <p>(f) Data Input and Output functions: Formatted I/O: printf(), scanf(). Character I/O format: getch(), getche(), getchar(), getc(), gets(), putchar(), putc(), puts().</p> <p>(g) Arrays: (One and two dimensional), declaring array variables, initialization of arrays, accessing array elements.</p> <p>(h) Strings: Declaring and initializing String variables, Character and string handling functions (strcpy, strcat, strchr, strcmp, strlen, strstr).</p> <p>(i) Iterations: Control statements for decision making: (a) Branching: if statement, if..else statement , else.. if statement, nested if statement, switch statement. (b) Looping: while loop, do.. while, for loop, nested loop. (c) Loop interruption statements: break, continue.</p> | 15 L |
| Unit II | <p>Functions, Pointers and Structures</p> <p>(a) Functions: Global and local variables, Function definition, return statement, calling a function.</p> <p>(b) Recursion: Definition, Recursion functions for factorial, Fibonacci sequence, exponential function, G.C.D.</p> <p>(c) Storage classes: Automatic variables, External variables, Static variables, Register variables.</p> <p>(d) Pointer: Fundamentals, Pointer variables, Referencing and de-referencing, Pointer Arithmetic, Pointers and Arrays, Array of Pointers, Pointers as function arguments.</p> <p>(e) Structure: Declaration of structure, reading and assignment of structure variables, Array of structures.</p> | 15 L |

Unit III Relational Database Management System

15 L

- (a) Introduction to Database Concepts:** Database, Overview of database management system. Database Languages- Data Definition Language (DDL) and Data Manipulation Languages (DML).
- (b) Entity Relation Model:** Entity, attributes, keys, relations, Designing ER diagram, integrity constraints over relations, Conversion of ER to relations with and without constraints.
- (c) SQL commands and Functions:**
 - (i) Creating and altering tables:** CREATE statement with constraints like KEY, CHECK, DEFAULT, ALTER and DROP statement.
 - (ii) Handling data using SQL:** selecting data using SELECT statement, FROM clause, WHERE clause, HAVING clause, ORDER BY, GROUP BY, DISTINCT and ALL predicates, Adding data with INSERT statement, changing data with UPDATE statement, removing data with DELETE statement.
 - (iii) Functions:** Aggregate functions-AVG, SUM, MIN, MAX and COUNT, Date functions- ADD_MONTHS(), CURRENT_DATE(), LAST_DAY(), MONTHS_BETWEEN(), NEXT_DAY(). String functions- LOWER(), UPPER(), LTRIM(), RTRIM(), TRIM(), INSTR(), RIGHT(), LEFT(), LENGTH(), SUBSTR(). Numeric functions: ABS(), EXP(), LOG(), SQRT(), POWER(), SIGN(), ROUND(number).
 - (iv) Joining tables:** Inner, outer and cross joins, union.

Unit IV Introduction to PL/SQL

15 L

- (a) Fundamentals of PL/SQL:** Defining variables and constants, PL/SQL expressions and comparisons: Logical Operators, Boolean Expressions, CASE Expressions Handling, Null Values in Comparisons and Conditional Statements,
- (b) PL/SQL Data types:** Number Types, Character Types, Boolean Type. Datetime and Interval Types.
- (c) Overview of PL/SQL Control Structures:** Conditional Control: IF and CASE Statements, IF-THEN Statement, IF-THEN-ELSE Statement, IF-THEN-ELSIF Statement, CASE Statement,
- (d) Iterative Control:** LOOP and EXIT Statements, WHILE-LOOP, FOR-LOOP, Sequential Control: GOTO and NULL Statements.

References:

- (a)** Programming in ANSI C (Third Edition) : E Balagurusamy, TMH
- (b)** Let us C by Yashwant Kanetkar, BPB.
- (c)** Database Management Systems, Ramakrishnam, Gehrke, McGraw-Hill
- (d)** Ivan Bayross, "SQL,PL/SQL -The Programming language of Oracle", B.P.B. Publications, 3rd Revised Edition.
- (e)** George Koch and Kevin Loney , ORACLE "The Complete Reference", Tata McGraw Hill, New Delhi.

Additional References:

- (a) Mastering Algorithms with C, Kyle Loudon, Shroff Publishers.
- (b) Algorithms in C (Third Edition): Robert Sedgewick , Pearson Education Asia.
- (c) Programming in ANSI C by Ram Kumar, Rakesh Agrawal, TMH.
- (d) Programming with C (Second Edition): Byron S Gottfried (Adapted by Jitender Kumar Chhabra) Schaum's Outlines (TMH)
- (e) Programming with C : K R Venugopal, Sudeep R Prasad TMH Outline Series.
- (f) Unix and C : M.P. Bhawe and S.A. Pateker, Nandu printers and publishers private limited.
- (g) Elsmasri and Navathe, "Fundamentals of Database Systems", Pearson Education.
- (h) Peter Rob and Coronel, "Database Systems, Design, Implementation and Management", Thomson Learning
- (i) C.J.Date, Longman, "Introduction to database Systems", Pearson Education.
- (j) Jeffrey D. Ullman, Jennifer Widom, "A First Course in Database Systems", Pearson Education.
- (k) Martin Gruber, "Understanding SQL", B.P.B. Publications.
- (l) Michael Abbey, Michael J. Corey, Ian Abramson, Oracle 8i – A Beginner's Guide, Tata McGraw-Hill.

| Course code | Topics for Practical | Credits |
|------------------|---|------------|
| USACCS5P1 | | Two |
| | <ol style="list-style-type: none">1. Write a C program that illustrates the concepts of C operators, mathematical functions, arrays and string.2. Write a C program that illustrates the concepts of different iterations.3. Write a C program that illustrates the concepts of functions, recursion and storage classes.4. Write a C program that illustrates the concepts of pointers and structure.5. Creating a single table with/ without constraints and executing queries. Queries containing aggregate, string and date functions fired on a single table.6. Updating tables, altering table structure and deleting table Creating and altering a single table and executing queries. Joining tables and processing queries.7. Writing PL/SQL Blocks with basic programming constructs.8. Writing PL/SQL Blocks with control structures. | |

| SEMESTER VI | | |
|-----------------|---|---------|
| Course code | Title | Credits |
| USACCS601 | Computer Programming and System Analysis –II | Two |
| Unit I | Object Oriented Systems | 15 L |
| | <p>(a) System: Information system, Purpose of system analysis and design, System Development Life Cycle.</p> <p>(b) Object-Oriented approach: Comparison between structured and object oriented approach. Features of object-orientations: Abstraction, Inheritance, Encapsulation and Polymorphism. Introduction to Object-Oriented Analysis and Design.</p> <p>(c) System Analysis: Overview of Unified Modeling Language (UML), Fact finding using User involvement: Actors, Use cases and Use case diagram.</p> <p>(d) Class diagram: Classes and Objects, Attributes and methods, Links and association: Links between objects, associations between classes, Ternary and reflexive associations, Role Names, aggregation, Generalization and inheritance.</p> | |
| Unit II | Introduction to Java Programming | 15 L |
| | <p>(a) Introduction: History of Java, Java features, different types of Java programs, Differentiate Java with C. Java Virtual Machine.</p> <p>(b) Java Basics: Variables and data types, declaring variables, literals: numeric, Boolean, character and string literals, keywords, type conversion and casting. Standard default values. Java Operators, Loops and Controls (No Questions are to be asked on this topic).</p> <p>(c) Classes: Defining a class, creating instance and class members: creating object of a class; accessing instance variables of a class; creating method; naming method of a class; accessing method of a class; overloading method; 'this' keyword, constructor and Finalizer: Basic Constructor; parameterized constructor; calling another constructor; finalize() method; overloading constructor.</p> <p>(d) Arrays: one and two-dimensional array, declaring array variables, creating array objects, accessing array elements.</p> <p>(e) Access control: public access, friendly access, protected access, private access.</p> | |
| Unit III | Inheritance, Exception Handling | 15 L |
| | <p>(a) Inheritance: Various types of inheritance, super and subclasses, keywords- 'extends'; 'super', overriding method, final and abstract class: final variables and methods; final classes, abstract methods and classes. Concept of interface.</p> <p>(b) Exception Handling and Packages: Need for Exception Handling, Exception Handling techniques: try and catch; multiple catch statements; finally block; usage of throw and throws. Concept of package. Integer class method: parseInt().</p> | |
| Unit IV | Java Applets and Graphics Programming | 15 L |
| | <p>(a) Applets: Difference of applet and application, creating applets, applet life cycle, passing parameters to applets.</p> | |

- (b) **Graphics, Fonts and Color:** The graphics class, painting, repainting and updating an applet, sizing graphics. Font class, draw graphical figures - lines and rectangle, circle and ellipse, drawing arcs, drawing polygons. Working with Colors: Color methods, setting the paint mode.
- (c) **AWT package:** Containers: Frame and Dialog classes, Components: Label; Button; Checkbox; TextField, TextArea.

References:

- (1) Programming with Java: A Primer 4th Edition by E. Balagurusamy, Tata McGraw Hill.
- (2) Java The Complete Reference, 8th Edition, Herbert Schildt, Tata McGraw Hill

Additional References:

- (1) Eric Jendrock, Jennifer Ball, D Carson and others, The Java EE 5 Tutorial, Pearson Education, Third Edition, 2003.
- (2) Ivan Bayross, Web Enabled Commercial Applications Development Using Java 2, BPB Publications, Revised Edition, 2006
- (3) Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD), Third Edition, 2004
- (4) The Java Tutorials of Sun Microsystems Inc. <http://docs.oracle.com/javase/tutorial>

| Course code | Topics for Practical | Credits |
|-------------|----------------------|---------|
| USACCS6P1 | | Two |

1. Write a Java program to create a Java class: (a) without instance variables and methods, (b) with instance variables and without methods, (c) without instance variables and with methods. (d) with instance variables and methods.
2. Write a Java program that illustrates the concepts of one, two dimension arrays.
3. Write a Java program that illustrates the concepts of Java class that includes (a) constructor with and without parameters (b) Overloading methods.
4. Write a Java program to demonstrate inheritance by creating suitable classes.
5. Write a program that illustrates the error handling using exception handling.
6. Write a program that demonstrates the use of input from the user using parseInt().
7. Write a Java applet to demonstrate graphics, Font and Color classes.
8. Write a Java program to illustrate AWT package.

Internal Assessment of Theory Core Courses Per Semester Per Course (Total 40 marks)

- (a) One Assignments: 10 Marks.
- (b) One Class Test:20 Marks.
- (c) Active participation in class instructional deliveries05 Marks.
- (d) Overall conduct as a responsible student, mannerism etc :05 Marks.

Semester End Theory Examination (Total 60 marks)

Theory: At the end of the semester, examination of two and half hours duration and 60 marks based on the four units shall be held for each course.

Pattern of **Theory question** paper at the end of the semester for **each course**:
There shall be Five compulsory Questions of 12 marks each with internal option.
Question 1 based on Unit I, Question 2 based on Unit II, Question 3 based on Unit III,
Question 4 based on Unit IV and Question 5 based on all four Units combined.

Semester End Practical Examination (Total 100 marks)

Semester V: Total evaluation is of 100 marks-

- (a) One Case study based on topics from semester V - 30 Marks
- (b) Viva based on Case study presented - 10 Marks
- (c) Semester End practical examination on computer - 50 marks.
- (d) Journal - 10 Marks.

Semester VI: Total evaluation is of 100 marks-

- (a) Project on any topic from semester V and/or VI - 30 Marks
- (b) Viva based on project submitted - 10 Marks
- (c) Semester End practical examination on computer - 50 marks.
- (d) Journal - 10 Marks.

1. There shall be TWO compulsory questions of TWENTY FIVE marks each with internal option for the semester end practical examination on computer.
2. The questions to be asked in the practical examination shall be from the list of practical experiments mentioned in the practical topics. A few simple modifications may be expected during the examination.
3. The semester end practical examination on the machine will be of THREE hours. In addition, TWO hours may be allotted for the Case study presentation of semester V/ project presentation of semester VI for a batch of students.
4. Students should carry a certified journal with minimum of 06 practicals (mentioned in the practical topics) at the time of examination.
5. Number of students per batch for the regular practical should not exceed 20. Not more than two students are allowed to do practical experiment on one computer at a time.

Guidelines for Case Study to be undertaken in Semester V

The syllabus proposes a case study to be done by students during semester V. The rationale behind introducing the case study is to motivate learner to choose a topic and explore the topic on a self study mode. It is expected that undertaking a case study would equip the learner with much needed understanding and confidence to do projects, which is proposed to be undertaken in the sixth semester.

- Topics for the case study can be chosen by the students from the list of topics given by the teacher.
- It is expected that a case study may be taken from C programming language, SQL or P/L SQL.
- Students are expected to study the topics by themselves and by referring books, internet links etc and make case study reports which cover different aspects of the topics discussed.

In two hours duration, students have to make a presentation of about 10 -15 minutes based on the case study. The examiners will evaluate the case study based on the following evaluation scheme:

- Content = 10 marks
- Understanding = 10 marks
- Documentation/Presentation = 10 marks

Total marks = 30

Guidelines for Project to be done in Semester VI

The syllabus proposes the introduction of a project to be done by students at semester VI of the applied component course. The objective of this step is to make learning more student- directed and see to it that the concepts covered in the syllabus be applied by the learner to real world situations. Working on a project is expected to help the learner to understand and appreciate the coverage of the topics discussed in the syllabus much better. This would also help the learner to face the industrial and professional demands (at least partially) once he or she completes the course.

- The projects shall be undertaken by the students under the guidance of the teacher teaching applied component courses or the experts approved by the teacher in charge.
- The whole class shall be divided into different batches, which can be distributed to different teachers teaching the applied component courses.
- Each student can chose a topic with the approval of the teacher in charge.
- The topic selected should be related to the topics covered in the syllabus.
- The project work should be spread over a period of at least 8 weeks.
- The project should cover problem solving using the concepts mentioned in the syllabus, approved by the teacher.
- Students may use the technology or programming languages covered in the syllabus. However, they may have the freedom to use other technologies or programming languages.
- Weightage shall be given to live projects and the projects with new concepts.
- At the end of the project, the students need to submit a typed project report of around 10- 15 pages detailing the following:
 - I. Title
 - II. Introduction
 - III. Objective
 - IV. Methodology
 - V. Experimental set up
 - VI. Results
 - VII. Conclusion
 - VIII. Reference
 - IX. Appendix (includes the coding used and additional results (if any))

In two hours duration, the students have to make a presentation of about 10 -15 minutes based on the project in front of the examiners. The examiners will evaluate the project based on the following evaluation scheme:

- Content = 10 marks
- Understanding = 10 marks
- Documentation/Presentation = 10 marks

Total marks = 30

Workload

Theory : 4 lectures per week .

Practicals: 2 practicals each of 2 lecture periods per week per batch. Two lecture periods of the practicals shall be conducted in succession together on a single day.
