

# SCHEME OF EXAMINATION AND DETAILED SYLLABUS

Faculty of CS & IT

**Master of Science  
(M.Sc. IT)**


(Duration- 2 Years)

(For 2019 Batch)



Contact us:

 8252299990

 8404884433



**AISECT University, Hazaribag**

Matwari Chowk, in front of Gandhi Maidan, Hazaribag (JHARKHAND)-825301

## MASTER OF SCIENCE (IT)

Duration: 24 Months (2 Years) Eligibility: B.Sc. (IT/CS/BCA), BE (CS/IT)

<b>COURSE STRUCTURE OF MSC ( IT )SEMESTER I</b>													
<b>Course Details</b>				<b>External Assessment</b>		<b>Internal Assessment</b>				<b>Credit Distribution</b>			<b>Allotted Credits</b>
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Total Marks</b>	<b>Major</b>		<b>Minor</b>		<b>Sessional</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>Subject wise Distribution</b>
				<b>Max Marks</b>	<b>Min Marks</b>	<b>Max Marks</b>	<b>Min Marks</b>	<b>Max Marks</b>	<b>Min Marks</b>				
<b>Theory Group</b>													
6IMIT101	<b>Core Course</b>	Fundamentals of Computers & MS Office	100	50	17	20	8	30	12	4	-	-	4
6IMIT102	<b>Core Course</b>	Programming Methodology & Programming in C	100	50	17	20	8	30	12	4	-	-	4
6IMIT103	<b>Core Course</b>	Computer Networks & Internet	100	50	17	20	8	30	12	4	-	-	4
6IMIT104	<b>Core Course</b>	Digital Electronics & Logic Design	100	50	17	20	8	30	12	4	-	-	4
<b>Practical Group</b>				<b>Term End Practical Exam</b>				<b>Sessional</b>					
6IMIT101	<b>Practical</b>	Fundamentals of Computers & MS Office	50	25	08	-	-	25	10	-	-	2	2
6IMIT102	<b>Practical</b>	Programming Methodology & Programming in C	50	25	08	-	-	25	10	-	-	2	2
<b>Grand Total</b>			<b>500</b>	-		-		-	-	-	-	-	<b>20</b>

Minimum Passing Marks are equivalent to Grade D

Major- Term End Theory Exam/ Practical Exam

Minor- Pre University Test

Sessional weightage – Attendance 50%, Three Class Tests/Assignments 50%

L- Lectures T- Tutorials P- Practical

## MASTER OF SCIENCE (IT)

Duration: 24 Months (2 Years) Eligibility: B.Sc. (IT/CS/BCA), BE (CS/IT)

<b>COURSE STRUCTURE OF MSC ( IT )SEMESTER II</b>													
<b>Course Details</b>				<b>External Assessment</b>		<b>Internal Assessment</b>				<b>Credit Distribution</b>			<b>Allotted Credits</b>
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Total Marks</b>	<b>Major</b>		<b>Minor</b>		<b>Sessional</b>		<b>L</b>	<b>T</b>	<b>P</b>	<b>Subject wise Distribution</b>
				<b>Max Marks</b>	<b>Min Marks</b>	<b>Max Marks</b>	<b>Min Marks</b>	<b>Max Marks</b>	<b>Min Marks</b>				
<b>Theory Group</b>													
6IMIT201	<b>Ability Enhancement</b>	Objects Oriented Programming with C++	100	50	17	20	8	30	12	4	-	-	4
6IMIT202	<b>Core Course</b>	DBMS & SQL	100	50	17	20	8	30	12	4	-	-	4
6IMIT203	<b>Core Course</b>	Data Structure	100	50	17	20	8	30	12	4	-	-	4
6IMIT204	<b>Core Course</b>	Operating Systems	100	50	17	20	8	30	12	4	-	-	4
<b>Practical Group</b>				<b>Term End Practical Exam</b>				<b>Sessional</b>					
6IMIT201	<b>Practical</b>	Objects Oriented Programming with C++	50	25	08	-	-	25	10	-	-	2	2
6IMIT202	<b>Practical</b>	DBMS & SQL	50	25	08	-	-	25	10	-	-	2	2
<b>Skill Courses</b>								<b>Sessional</b>					
6SCMIT201	<b>Skill Enhancement</b>	Skill Enhancement Elective Course I	50	-	-	-	-	50	20	1	-	1	2
<b>Grand Total</b>			<b>550</b>	-		-		-	-	-	-	-	<b>22</b>

Minimum Passing Marks are equivalent to Grade D

Major- Term End Theory Exam / Practical Exam

Minor- Pre University Test

Sessional weightage – Attendance 50%, Three Class Tests/Assignments 50%

Skill Elective I– Any other course being offered in this semester as per the list given at the end of course structure.

L- Lectures T- Tutorials P- Practical

## MASTER OF SCIENCE (IT)

Duration: 24 Months (2 Years) Eligibility: B.Sc. (IT/CS/BCA), BE (CS/IT)

<b>COURSE STRUCTURE OF MSC ( IT ) SEMESTER III</b>													
<b>Course Details</b>				<b>External Assessment</b>		<b>Internal Assessment</b>				<b>Credit Distribution</b>			<b>Allotted Credits</b>
Course Code	Course Type	Course Title	Total Marks	Major		Minor		Sessional		L	T	P	Subject wise Distribution
				Max Marks	Min Marks	Max Marks	Min Marks	Max Marks	Min Marks				
<b>Theory Group</b>													
6IMIT301	<b>Ability Enhancement</b>	Programming with VB.Net	100	50	17	20	8	30	12	4	-	-	4
6IMIT302	<b>Core Course</b>	Java Programming	100	50	17	20	8	30	12	4	-	-	4
6IMIT303	<b>Core Course</b>	(A)Software Engineering, (B) Computer Architecture	100	50	17	20	8	30	12	4	-	-	4
6IMIT304	<b>Core Course</b>	A. Compiler Design, B. Artificial Intelligence and Expert System, C. Data warehousing & Mining	100	50	17	20	8	30	12	4	-	-	4
<b>Practical Group</b>				<b>Term End Practical Exam</b>				<b>Sessional</b>					
6IMIT301	<b>Practical</b>	Programming with VB.Net	50	25	08	-	-	25	10	-	-	2	2
6IMIT302	<b>Practical</b>	Java Programming	50	25	08	-	-	25	10	-	-	2	2
<b>Skill Courses</b>								<b>Sessional</b>					
6SCITM301	<b>Skill Enhancement</b>	Skill Enhancement Elective Course II	50	-	-	-	-	50	20	1	-	1	2
<b>Grand Total</b>			<b>550</b>	-	-	-	-	-	-	-	-	-	<b>22</b>

Minimum Passing Marks are equivalent to Grade D  
Major- Term End Theory Exam/ Practical Exam

L- Lectures T- Tutorials P- Practical

## MASTER OF SCIENCE (IT)

Duration: 24 Months (2 Years) Eligibility: B.Sc. (IT/CS/BCA), BE (CS/IT)

### COURSE STRUCTURE OF MSC ( IT ) SEMESTER IV

Course Details				External Assessment		Internal Assessment				Credit Distribution			Allotted Credits
Course Code	Course Type	Course Title	Total Marks	Major		Minor		Sessional		L	T	P	Subject wise Distribution
				Max Marks	Min Marks	Max Marks	Min Marks	Max Marks	Min Marks				
<b>Theory Group</b>													
6IMIT401	<b>Core Course</b>	Soft Computing	100	50	17	20	8	30	12	4	-	-	4
6IMIT402	<b>Core Course</b>	Advanced Java Programming	100	50	17	20	8	30	12	4	-	-	4
6IMMA401	<b>Core Course</b>	Numerical Methods	100	50	17	20	8	30	12	4	-	-	4
<b>Practical Group</b>				<b>Term End Practical Exam</b>			<b>Sessional</b>						
6IMIT402	<b>Practical</b>	Advanced Java Programming	50	25	08	-	-	25	10	-	-	2	2
6IMIT403	<b>Practical</b>	Major Project	150	75	25	-	-	75	30	-	-	6	6
<b>Grand Total</b>			<b>500</b>	-		-		-	-	-	-	-	<b>20</b>

Minimum Passing Marks are equivalent to Grade D

Major- Term End Theory Exam / Practical Exam

Minor- Pre University Test

Sessional weightage – Attendance 50%, Three Class Tests/Assignments 50%

Compulsory Project/Dessertation with choice in any Disciplinary specific elective. Compulsory one paper presentation certificate in related dicipline.

L- Lectures T- Tutorials P- Practical

**SKILL ENHANCEMENT ELECTIVE COURSES**

<b>Non-Technical</b>			
<b>Elective No.</b>	<b>Department/ Faculty Name</b>		
	<b>Faculty of Information Technology</b>		
I	SCIT 201	Data Entry Operation	2(1+0+1)
II	SCIT 301	Multimedia	2(1+0+1)
III	SCIT 501	Web Designing with HTML	2(1+0+1)
IV	SCMIT 201	Web Development	2(1+0+1)
V	SCMIT 301	LINUX	2(1+0+1)
	<b>Faculty of Management</b>		
I	SMGT 201	Briefing and Presentation Skills	2(1+0+1)
II	SMGT 301	Resolving Conflicts and Negotiation Skills	2(1+0+1)
III	SMGT 802	Entrepreneurship Development	2(1+0+1)
	<b>Faculty of Commerce</b>		
I	SCOM 201	Tally ERP 9	2(1+0+1)
II	SCOM 302	Multimedia	2(1+0+1)
III	SCOM 803	Data Analyst	2(1+0+1)
	<b>Faculty of Humanities</b>		
I	SHBA 301	Pursuing Happiness	2(1+0+1)
II	SHBA302	Communication Skill and Personality Development	2(1+0+1)
III	SHMA301	Tourism in M.P	2(1+0+1)
	<b>Faculty of Science</b>		
I	SSBI 301	Mushroom Cultivation	2(1+0+1)
II	SSPH 301	House Hold Wiring	2(1+0+1)
III	SSPH 301	Basic Instrumentation	2(1+0+1)
IV	SSPH 301	DTP Operator	2(1+0+1)
V	SSCH 301	Graphic Designing	2(1+0+1)
	<b>Faculty of Education</b>		
I	SCBE 403	Understanding of ICTC (Information Communication Technology)	2(1+0+1)
II	SCPE 201	Yoga Education	2(1+0+1)

## **Program Objective:**

The objective of the Program is to develop skilled manpower in the various areas of information technology like:

1. To prepare graduates who will be successful professionals in industry, government, academia, research, entrepreneurial pursuit and consulting firms. Students will demonstrate ability to conduct a research or applied Computer Science project, requiring writing and presentation skills which exemplify scholarly style in computer science.
2. To prepare graduates who will contribute to society as broadly educated, expressive, ethical and responsible citizens with proven expertise
3. To prepare graduates who will achieve peer-recognition; as an individual or in a team; through demonstration of good analytical, design and implementation skills
4. To prepare graduates who will thrive to pursue life-long learning to fulfill their goals. Students will demonstrate a breadth of knowledge in computer science, as exemplified in the areas of systems, theory and software development.

## **Program Outcome:**

1. An ability to use current techniques, skills, and tools necessary for computing practices.
2. An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices;
3. An ability to apply design and development principles in the construction of software systems of varying complexity.



## **Program Specific Outcomes MSC**

The objectives are that, upon completion, the students will have developed:

1. The ability to gather, organise and critically evaluate information needed to formulate and solve problems;
2. The ability to apply acquired knowledge effectively and efficiently to all work in the relevant areas of Engineering;
3. Skills in qualitative and quantitative oral and written communications designed to present essential scientific and technical concepts;
4. Abilities in observation, measurement and the design and conduct of experiments through practical experience in the laboratory;
5. The ability to display initiative and innovation, demonstrated in a major item of individual work;
6. The ability to work collaboratively with others through the development of team skills;
7. The ability to apply advanced techniques to problems at the frontiers of knowledge;
8. A range of professional and management skills;
9. A deep knowledge and advanced understanding in specialist areas;
10. An awareness of research methodology.

## Course Outcomes of MSc(IT)

SUB CODE/ SUBJECT NAME	OUTCOMES
6IMIT101 Fundamentals of Computers & MS Office	<ul style="list-style-type: none"> <li>• An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.</li> <li>• An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.</li> <li>• An ability to analyze the local and global impact of computing on individuals, organizations, and society.</li> <li>• Demonstrate the basic mechanics of creating Word documents, presentation and excel calculation for office use.</li> </ul>
6IMIT102 Programming Methodology & Programming in C	After learning this paper, the student would be able to understand the concept of programming language, all about C language, its features, uses and working principles.
6IMIT103 Computer Networks & Internet	After study this student will be able to know about <ol style="list-style-type: none"> <li>1. Independently understand basic computer network technology.</li> <li>2. Understand and explain Data Communications System and its components.</li> <li>3. Identify the different types of network topologies and protocols.</li> <li>4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.</li> </ol>
6IMIT104 Digital Electronics & Logic Design	After study this student will be able to know about <ol style="list-style-type: none"> <li>1. Knowledge in the field of solid state materials.</li> <li>2. To analyze the structure of different types of semiconductor crystal structures. Know the intrinsic property of semiconductor materials.</li> <li>3. Idea about the equilibrium and non-equilibrium states of semiconductors.</li> <li>4. The complete internal structure of PN junction including different.</li> </ol>
6IMIT201 Objects Oriented Programming with C++	After study this student will be able to know about the concepts of OOPs with C++ language, classes. Student will be able to create Arrays its uses, uses of function overloading, inheritance & C++I/O system.
6IMIT202 DBMS & SQL	After study this student will be able to know about and concepts & fundamentals of DBMS, Concept of keys, RELATIONAL DATA MODEL& design. Student will also able to create table and implement commands.
6IMIT203 Data Structure	After study this student will be able to know about the concepts of Data Structure Using C++ Language, List & Its Operations Concept Of Tree, Algorithm& Graphs Design.
6IMIT204 Operating Systems	After study this student will be able to know about the functioning of Operating System. To make students able to learn different types of operating systems along with concept of file systems and CPU scheduling algorithms used in operating system. To provide students knowledge of memory management and deadlock handling algorithms. At the end of the course, students will be able to implement various algorithms required for management, scheduling, allocation and communication used in operating system.
6SCMIT201 – Web Development	After study this student will be able to understand the concept of Web Application Design and implementation. They will be able to identify the

		reason and importance of web application development and design.
6IMIT301	Programming with VB.Net	<p>After the completion of the course, students are expected to:</p> <ol style="list-style-type: none"> <li>1. Have gained a good understanding of the basic concepts of object orientation</li> <li>2. Have a good understanding of the visual basic language structure and language syntax</li> <li>3. Have developed the ability to design and develop interactive applications using the object-oriented principals, encapsulation, inheritance and to some extents polymorphism</li> <li>4. Be able to effectively develop applications with full functionality and a graphical user interface using the language visual basic</li> <li>5. Have the capability of analysing and finding suitable and effective solutions to windows based applications using classes and objects</li> </ol>
6IMIT302	Java Programming	<p>Students will complete software projects comprised of an object-oriented design, implementation, and test plan.</p> <ul style="list-style-type: none"> <li>• Designs will demonstrate the use of good object-oriented design principles including encapsulation and information hiding.</li> <li>• The implementation will demonstrate the use of a variety of basic control structures including selection and repetition; classes and objects in a tiered architecture (user interface, controller, and application logic layers); primitive and reference data types including composition; basic AWT components; file-based I/O; and one-dimensional arrays.</li> <li>• Test plans will include test cases demonstrating both black box and glass box testing strategies.</li> </ul>
6IMIT303 (A)	Software Engineering	<p>After study this student will be able to</p> <p>Carry out an evaluation and selection of projects against strategic, technical and economic criteria and use a variety of cost benefit evaluation techniques for choosing among competing project proposals. Approach project planning in an organized step by step manner and select an appropriate process model produce an activity plan for a project.</p> <ul style="list-style-type: none"> <li>• Identify project risks, monitor and track project deadlines and produce a work plan and resource schedule.</li> <li>• Plan the evaluation of a proposal or a product and manage people in software environments. Understand the importance of teamwork and quality management in software project management. Apply these project management tools and techniques in a diversity of fields such as new product and process development, construction, information technology, health care, and applied research.</li> </ul>
6IMIT303(B)	Computer Architecture	<p>After study this student will be able to know about the basic computer organization, design and micro-operations, Understanding of CPU functioning and computer arithmetic. Learning various methods and techniques of memory organization.</p>
6IMIT304	Compiler Design - A	<ol style="list-style-type: none"> <li>1. To apply the knowledge of lex tool &amp; yacc tool to develop a scanner &amp; parser.</li> <li>2. To design &amp; conduct experiments for Intermediate Code Generation in compiler.</li> <li>3. To design &amp; implement a software system for backend of the compiler.</li> <li>4. To deal with different translators.</li> <li>5. To develop program to solve complex problems in compiler</li> <li>6. To learn the new code optimization techniques to improve the performance of a program in terms of speed &amp; space.</li> </ol>

	<p>7. To acquire the knowledge of modern compiler &amp; its features.</p> <p>8. To learn &amp; use the new tools and technologies used for designing a compiler</p>
6IMIT304 B. Artificial Intelligence and Expert System	After study this student will be able to know about the AI with search algorithm and expert system with life cycle.
6IMIT304C. Data warehousing & Mining	<p>After study this student will be able to know about the</p> <ul style="list-style-type: none"> <li>• Process raw data to make it suitable for various data mining algorithms.</li> <li>• Discover and measure interesting patterns from different kinds of databases.</li> <li>• Apply the techniques of clustering, classification, association finding, feature selection and visualization to real world data.</li> </ul>
6SCITM301 Linux	After Study This Student Will Be Able To Know About Basic Features, Different flavors of Linux. Advantages, Installing. Student will know about Processes in Linux, Shell programming & Gnome graphical interfaces.
6IMIT401 Soft Computing	<ul style="list-style-type: none"> <li>• Fuzzy logic and its applications.</li> <li>• Artificial neural networks and its applications.</li> <li>• Solving single-objective optimization problems using GAs.</li> <li>• Solving multi-objective optimization problems using Evolutionary algorithms (MOEAs).</li> <li>• Applications of Soft computing to solve problems in varieties of application domains.</li> </ul>
6IMIT402 Advanced Java Programming	<p>Students will complete software projects comprised of an object-oriented design, implementation, and test plan.</p> <ul style="list-style-type: none"> <li>• Designs will demonstrate the use of good object-oriented design principles including encapsulation and information hiding.</li> <li>• The implementation will demonstrate the use of a variety of basic control structures including selection and repetition; classes and objects in a tiered architecture (user interface, controller, and application logic layers); primitive and reference data types including composition; basic AWT components; file-based I/O; and one-dimensional arrays.</li> <li>• Test plans will include test cases demonstrating both black box and glass box testing strategies.</li> </ul>
6IMMA401 Numerical Methods	<p>1) Knowledge and Understanding: Students are able to understand the nature and operations of Numerical Analysis, demonstrate familiarity with theories and concepts used in Numerical Analysis and identify the steps required to carry out a piece of research on a topic in Numerical Analysis, derivation of the Numerical Methods, studying their convergence rate and performance, applicability of the methods on different test examples.</p> <p>2) Intellectual Skills: By the end of the course the student is expected to solve real-life and Engineering applications reflecting the student ability to:</p> <ul style="list-style-type: none"> <li>• Recognize and apply appropriate theories, principles and concepts relevant to Numerical Analysis.</li> <li>• B2 Critically assess and evaluate the literature within the field of Numerical Analysis.</li> <li>• B3 Analyze and interpret information from a variety of sources relevant to Numerical Analysis.</li> </ul> <p>3) Practical Skills: By the end of the course student will have the ability to compare the computational methods for advantages and drawback, choose the suitable computational method among several existing methods ,</p>

implement the computational methods using any of existing programming languages, test such methods and compare between them, identify the suitable computational technique for a specific type of problems and develop the computational method that is suitable for the underlying problem.

- 4) Transferable Skills: Within the lectures the student is able to transfer ideas and experience Numerical Analysis Techniques, work effectively both in a team and independently, apply the best computational methods to solve real-life and Engineering applications via computational packages such as MATLAB or Mathematica and develop his ability to self appraise and reflect on practice relevant to Numerical Analysis.

## FUNDAMENTALS OF COMPUTERS & MS OFFICE

### COURSE OBJECTIVE:

Student will be able

1. To understand the basic knowledge of computer
2. To understand the Number System
3. To understand the Computer Virus and Internet
4. To understand the concept of Storage devices.
5. To understand about software as well as hardware.
6. To understand the basic knowledge of MS Windows.
7. To understand the Office Packages.
8. To understand the MS Excel.

### Syllabus:

#### UNIT-I

**Introduction** - Introduction, Limitations of computers, Fundamental uses of computers, Types of Computers, Generations of Computers.

**Personal Computer** - Introduction, Components of personal computers, Evolution of PCs.

**Boolean Algebra and Logic Gates** - Introduction, Boolean Algebra And Operator, OR Operator, NOT Operator, Basic Postulates of Boolean Algebra, Basic Logic Gates.

**Number System** - Introduction, Digital and Analog Operations, Binary Data, Binary Number System, Decimal Number System, Octal Number System, Hexadecimal Number System, Coding System.

#### UNIT-II

**Data Representation and Binary Arithmetic** - Introduction, Bits, Nibbles, Bytes and Words, Data Representation, Coding system, Binary Arithmetic, Binary Addition, Binary Subtraction, Binary Multiplication, Binary Division.

**Input Devices** - Introduction, Input Device,

**Output Devices** - Introduction, Output Devices, Soft Copy Vs Hard Copy Output, Monitor, Printers, Plotter.

**Central Processing Unit** - Introduction, What is Central Processing Unit, Arithmetic And Logic Unit, Control Unit, Registers, Instruction set, Processor Speed.

**Storage Devices** - Introduction, Storage and its needs, Primary Storage, Secondary Storage.

#### UNIT-III

**Basics of Software-** Introduction, What Does Software Stand For ?, Needs of software, Types of software, Open Source Software,

**Operating System** - Introduction, Operating System, Why an Operating System, Functions of Operating System, the Booting Process, Types of Reboot,

**Disk Operating System** - Introduction, What is DOS?, Functions of DOS, Versions of DOS, DOS Commands , Important Internal Commands of DOS, Important External Commands of dos, Executable Vs Non-Executable Files In Dos

**Programming Languages** - Introduction, Data, information And Knowledge, Characteristics of Information, What is a program?, What is a Programming language?, Programming approaches, Types of Programming Language.

## UNIT-IV

**Computer Virus** - Introduction, Virus, History, Mechanism of virus, Types of Computer Virus, Related Concepts: Anti Virus Programs,

**Communication and IT** - Introduction, Computer Network, Communication Process, Communication Types, Transmission Media, Wireless Media, Communication Channels/Media, Modem, Characteristics of a Modem, Types of Modem

**Networks** - Introduction, Internet Vs Intranet, Types of Network, Topology, Types of Connectivity, Network Devices.

## UNIT-V

**Know the Windows Operating System** - Introduction, What is Windows XP ?, Evolution of Windows Operating System, Features of Windows XP, What's New in Windows XP, Windows and Its Elements, Accessories, Files and Folders

**Microsoft Office Package** – Introduction about MS Word ,Ms Excel, Ms PowerPoint.

## Practical's

1. Creating, opening, closing, saving and editing a word Document..
2. Insertion of header and footer in the document.
3. Use of word art, spell check and work with Page layout.
4. Creation of a link between two files using Hyperlink.
5. E- mail-merge and providing protection of a document.
6. Creation of a letter/Application in different subjects.
7. How to insert, close, update and save a worksheet?
8. Creation of records in excels for students marks of five subjects and calculation of their average percentage using formulas.
9. Operation of data sorting in a worksheet.
10. Use of mathematical functions, date function and time function.
11. Define trig function with an operation on excel sheet.
12. Creation of new slide and duplicate slide in power point.
13. Steps of presentation and creation of presentation for the seminar in a topic.
14. Use of animation audio and clipart in power point presentation.
15. Changing backgrounds and adding slides in a presentation

## COURSE OUTCOMES:-

- An ability to apply knowledge of computing and mathematics appropriate to the program's student outcomes and to the discipline.
- An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution.
- An ability to analyze the local and global impact of computing on individuals, organizations, and society.
- Demonstrate the basic mechanics of creating Word documents ,presentation and excel calculation for office use.

## Text Books:

- Computer Fundamental (3rd Ed) Sinha,P.K.
- Fundamental of Information Technology ShritvastavaCheton
- Fundamentals of Computers, Murthy,C.S.V. Delhi S. K. Kataria& Sons.
- MS office XP for Everyone, Saxena Sanjay, New Delhi Vikas Publication

**Reference Books:**

- Computer fundamental: V .Rajaraman; PHI
- Fundamental Of IT Leon and Leon Leon Tec World
- Fundamental of Computer Programming & Information Technology Sharma,G.& Singh G.  
Delhi S. K. Kataria& Sons

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**



## PROGRAMMING METHODOLOGY & PROGRAMMING IN C

### COURSE OBJECTIVE:

Student will be able

1. To understand the basic knowledge of programming concepts.
2. To understand the C language & its concepts.
3. To understand the basic concept of programming and develop the programming skills.

### Syllabus:

#### UNIT - I

**Principles of Programming**, Introduction to Programming, Program Concept, Characteristics of Programming, Stages in Program Development, Tips for Program Designing, Programming Aids, Algorithms, Notations, Design, Flowcharts, Symbols, Rules

**Programming Techniques and Logic**, Introduction, Introduction to programming techniques, Top-down approach or technique, Bottom-up approach or technique, Unstructured technique of programming, Structured technique of programming, Modular technique of programming, Comparative study of programming techniques, Debugging , Syntax Errors, Logical Errors, Data Entry Errors, Linker Errors, Runtime Errors, Program Testing

#### UNIT - II

**Introduction to 'C'**, Introduction, Structure of a C program, 'C' Tokens, Keywords, Identifiers, 'C' Constants, Variables in C, Data Types, Derived Data Types : , Operators, Precedence and Associativity of operators, Hierarchy of operators at a glance, Expression & its Evolution, Type conversion in expressions , (Implicit and Explicit type conversion)

**Decision Making and Branching**, Introduction, Sequential statements, Unformatted I/O functions, Formatted input using scanf() function, Formatted output using print(), Branching statements, The if-else statement, The nested if-statement, The switch statement, Additional programs

#### UNIT - III

**Looping Statements**, Introduction, for-statement, while-statement, do-while statement, Difference between while-loop and do-while loop, Nested loops, Jumps in loops, Programming examples

**Arrays**, Introduction, Single-dimensional arrays, Reading and writing single dimensional arrays, Examples of Complex Programs, Searching, Sorting, Two-dimensional arrays (Multi-dimensional arrays), Reading-writing two-dimensional arrays, Manipulation in two-dimensional arrays, Programming Examples

**Strings**, Concepts of string, Strings in C language, String variable, Initializing strings, String input/output functions, Arrays of strings, String handling functions, Memory formatting

#### UNIT - IV

**User Defined Functions**, Introduction, Elements of user-defined functions, Categories of functions, Passing parameters to functions, Programming Examples, Arrays in functions, Nesting of Functions, Recursion, Command Line Arguments , Storage Classes

**Structure and Union**, Introduction to structures, Structure and its definition, Structure declaration, Tagged Structure, Structure variables, Type-Defined Structure, Structure initialization, Accessing structures, Nested structures, Array of structures, Structures and functions, Sending individual

members, Sending the whole structure, Passing structures through pointers, Uses of structures, Union and its definition

**Pointers**, Introduction, Pointer concepts, Pointer variable, Accessing variables through pointers, Pointer declaration and Definition, Initializing a pointer variable, Pointers to Pointers, Compatibility, Pointer applications, Pointers and other operators, Memory allocation functions, Memory map of C program, Memory management functions

## **UNIT - V**

**File Handling**, Introduction to file handling, File system basics, Standard streams in C, File structure, FILE pointer, Opening and closing a file, File handling functions, File types, Text and Binary, Input / Output operations on file, Reading a character using getc(), Writing a character using putc(), Using feof(), Working with string using fputs() and fgets(), Using fprintf() and fscanf(), Using fread() and fwrite(), Direct Access file, fseek()

### **Practical's:**

1. Write a C Program to add two integer numbers.
2. Write a C Program to Check Whether a Number is Even or Odd.
3. Write a C Program to Check Whether a Number is Positive or Negative or Zero.
4. Write a C Program to Display Fibonacci Series.
5. Write a C Program to Reverse a Number.
6. Write a C Program to Check Whether a Number is Palindrome or Not.
7. Write a C Program to Make a Simple Calculator to Add, Subtract, Multiply or Divide Using switch case.
8. Write a C Program to Calculate Factorial of a Number Using Recursion.
9. Write a C Program to Calculate Average Using Arrays.
10. Write a C Program to Add Two Matrix Using Multi-dimensional Arrays.
11. Write a C Program to Swap Numbers in Cyclic Order Using Call by Reference.

### **COURSE OUTCOME:**

After learning this paper, the student would be able to understand the concept of programming language, all about C language, its features, uses and working principles.

### **Reference Books:**

- , "Programming In C ", E. Balaguruswamy TMH Publications
- " Programming With C ", Gottfried, Schaum's Outline Series, TMH Publications
- " program design ", Peter Juliff, PHI Publications
- "Let us c", Yashwantkanetkar,

### **Text Books:**

- " Thinking In C ", Mahapatra, PHI Publications
- "Introduction To Computers And Information Technology ", AnuragSeetha, Ram Prasad & Sons, Bhopal.
- " Computers Today ", S.K.Basandra, Galgotia Publications.

## COMPUTER NETWORKS & INTERNET

### COURSE OBJECTIVE:

Student will be able

1. To understand the fundamental concepts of computer networking.
2. To understand the basic taxonomy and terminology of the computer networking area.
3. To understand the advanced networking concepts, preparing the student for entry Advanced courses in computer networking.
4. To understand the various transition method.

### Syllabus:

#### UNIT - I

**INTRODUCTION TO COMPUTER NETWORK**, Network, Computer Networks, Need of Network, Uses of Computer Network, Applications of networks, Network Criteria, Network Hardware and Software, network types : client, server & peers, Classification of Computer Network, Server, ATM (Asynchronous Transfer Mode), Modem

**THE THEORETICAL NETWORK MODEL - OSI**, OSI Model, open system interconnection model (OSI), Layered Architecture of the OSI Reference Model, Functions of the ISO/OSI Layers, Summary of OSI Layer functions

#### UNIT - II

**TRANSMISSION TECHNOLOGY**, transmission technology, Data can be analog or digital, Analog and Digital Transmission, asynchronous & synchronous transmission, Types of Communication Modes, BaseBand and Broadband Transmission, Comparison of Baseband and Broadband Signaling

**NETWORK TOPOLOGY**, Network Topology, Types of Network, Local Area Network (LAN), Metropolitan Area Networks (MAN), Wide Area Networks (WAN), Satellite Networks, Wireless LAN

#### UNIT - III

**TRANSMISSION MEDIA**, Transmission Media, Classification of Transmission Media, Comparison of Guided and Unguided Media, Twisted Pair (TP) Cable, Coaxial Cable, Fiber Optic Cable (FOC), Unguided Media, Radio Frequency Characteristics, Microwave Transmission, Applications of Infrared Transmission, Switching Methods, Packet switching, Circuit Switching, Message Switching

**DATA LINK LAYER**, Data Link Layer Design issues, Sliding Window Protocols

#### UNIT - IV

**NETWORK ADAPTERS**, Multiple Access Protocol, ALOHA, Carrier Sense Multiple Access (CSMA), CSMA/CD [Carrier Sense Multiple Access/Collision Detection], Collision Free Protocols, Limited Contention Protocol

**NETWORK LAYER**, Functions of Network Layer, Routing Algorithms, Congestion Control Algorithm

**APPLICATION LAYER**, Domain Name System, Simple Mail Transfer Protocol (SMTP), Hyper Text Transfer Protocol (HTTP), File Transfer Protocol

#### UNIT - V

**CRYPTOGRAPHY**, Encryption & Decryption - Cryptography, Terminology, Classification of Cryptography :, Substitution Ciphers :, Security of algorithms :, Steganography :, Steganography vs Cryptography :, public key encryption , Comparison of Symmetric and Asymmetric Key Cryptography ,

## **COURSE CODE: 6IMIT103**

Public Key Cryptanalysis, Digital Signature, Requirements of Digital Signature, Direct Digital Signature, Arbitrated Digital Signature, Authentication Protocols, Symmetric Encryption Approach, Public-Key Encryption Approach, Digital Signature Standard, RSA and Digital Signature, DSS Approach, The Digital Signature Algorithm

### **COURSE OUTCOME:**

After study this student will be able to know about

1. Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.

### **Reference Books:**

- “Networking Essentials:Study Guide MCSE”, Second Edition, James Chellis Charles Perkins, Matthew Strebe BPB Publications.
- MCSE Windows 2000 Network Infrastructure Disign
- “Data and Computer Communication” William Stallings
- Computer Networks Andrew S. Tanenbaum & David J. Wetherall

### **Text Books:**

- “Computer Network ” Andrew & Tanenbaum,
- “Local Area Networks”, S.K.Basandra & S. Jaiswal, Galgotia Publications

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## DIGITAL ELECTRONICS & LOGIC DESIGN

### COURSE OBJECTIVE:

Student will be able

1. To understand the overview of the principles, operation and application of the analog building blocks like diodes, BJT, FET etc for performing various functions.
2. To understand the overview of amplifiers, feedback amplifiers and oscillators.
3. To understand the knowledge on existing on future analog circuits, Registers, Circuits

### Syllabus:

#### UNIT - I

**Digital Number System** - Data Type, Number System, Decimal Number System, Binary Number System, Octal Number System, Hexadecimal Number System, Conversions, Decimal to Binary Number System, Decimal to Octal Number System, Decimal to Hexadecimal Number System, Binary-To-Decimal Conversion, Binary to Octal Number System, Binary to Hexadecimal Number System, Octal to Decimal Number System, Octal to Binary Number System, Octal to Hexadecimal Number System, Hexadecimal to Decimal Number System, Hexadecimal to Binary Number System, Hexadecimal to Octal Number System

**Binary Arithmetic** - Addition, Binary Number Addition, Octal Number Addition, Hexadecimal Number Addition, Subtraction, Binary Number Subtraction, Octal Number Subtraction, Hexadecimal Number Subtraction, Binary Multiplication, Binary Division, 1's Compliment, 2's Compliment, Subtraction Using 2's Compliment

#### UNIT - II

**Arithmetic Operations** - Integer Data, 8 Bit Signed Binary Representation, 16 Bit Signed Binary Representation, Fixed Point Representation, Floating Point Representation, Binary Codes, Weighted Code (8421, 2421, 528), Un Weighted Code, Excess 3 Code, Gray Code, Error Detection Code, Even Parity, Odd Parity, Error Detection and Correction Code, Alpha Numeric Code, EBCDIC

**Boolean Algebra** - Introduction, Boolean Postulates : , Boolean Identities :, OR Gate Identity, AND Gate Identity, NOT Gate Identity, Boolean Laws, Commutative Law, Associative Law, Distributive Law, Duality, DeMorgan's Theorem

**Logic Gates** - Logic Gates, NOT Gate, AND Gate, OR Gate, NAND Gate, NOR Gate, Ex OR Gate, Ex NOR Gate, Universal Gate

**Karnaugh Map** - Introduction, Karnaugh Map, MIN and MAX Terms, MIN and MAX Terms from Truth Table, Product of Sum (POS) Equation, Simplification of Boolean Equation by K Map, Two Variable K Map, Three Variable K Map, Four Variable K Map, Don't Care Condition

#### UNIT - III

**Combinational Circuits** - Introduction, Arithmetic Circuits, Adders, Half Adder, Full Adder, Serial Adder, Parallel Adder, BCD Adder, Subtractors, Half Subtractor, Full Subtractor, Parity Generator - Checker, Three Bit Parity Generator, Three Bit Parity Checker, Binary Comparators, Two Bit Comparator, Multiplexers, 4-to-1 Multiplexer, 16-to-1 Multiplexer, De-multiplexer, 1-to-4 De-

multiplexer, 1-to-16 De-multiplexer, Encoder, Decimal to BCD Encoder, Octal to Binary Encoder, Decoder  
**Sequential Circuits** - Introduction, Clock, FLIP FLOP, Latch, S-R Flip Flop, J-K Flip Flop, Master Slave J-K Flip Flop, Preset and Clear Inputs, T(Toggle) Flip Flop, D(Delay) Flip Flop, State Diagrams of Flip-Flops, State Diagram of SR Flip-Flop, State Diagram of JK Flip-Flop, State Diagram of T Flip-Flop, State

**COURSE CODE: 6IMIT104**

Diagram of D Flip-Flop, 8.5 Excitation Table of Flip-Flops, Excitation Table of SR Flip—Flop, Excitation Table of JK Flip-Flop, Excitation Table of D-Flip-Flop, Excitation Table of T-Flip-Flop

**UNIT - IV**

**Registers** - Introduction, Bit Shift Register, Serial In Serial Out (SISO), Serial In Parallel Out (SIPO), Parallel In Serial Out (PISO), Parallel In Parallel Out (PIPO), 4 Bit Left Shift Register, 4 Bit Right Shift Register, Universal Shift Register, Applications of Registers, Ring Counter, Self Correcting Ring Counter, Johnson or Twisted Ring Counter

**UNIT - V**

**Counters** - Introduction, 2-Bit Asynchronous Up Counter, 4-Bit Asynchronous Up Counter, 4-Bit Asynchronous Down Counter, Asynchronous Binary Up/Down Counter, Asynchronous Counter Design, 2-Bit Synchronous Counter, 3-Bit Synchronous Counter, 3-Bit Synchronous Down Counter, 3-Bit Synchronous Up/Down Counter, Design of Modulus N Synchronous Counters, Skipping State Counter, State Reduction and State Assignment, Comparison Between Asynchronous Vs Synchronous Counters

**Register Transfer Level (RTL)** - Introduction, Micro-operations, Transfer Micro-operations, Arithmetic Micro-operations, Logic Micro-operations, Shift Micro-operations, Instruction Code and Code Format, Design of a Simple Computer, Components Of Digital Computer, Designing a Simple Computer, Steps to Design a Computer

**COURSE OUTCOME:**

After study this student will be able to know about

1. Knowledge in the field of solid state materials.
2. To analyze the structure of different types of semiconductor crystal structures. Know the intrinsic property of semiconductor materials.
3. Idea about the equilibrium and non equilibrium states of semiconductors.
4. The complete internal structure of PN junction including different.

**Reference Books:**

- Computer System Architecture by: Morris Mano.
- Digital Computer Fundamentals by Bartee.

**Text Books:**

- Digital Computer Electronics by Malvino.
- Digital Computer Organization by Kamal Prakashan
- Modern Digital Electronics by R.P. Jain

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## OBJECTS ORIENTED PROGRAMMING WITH C++

### COURSE OBJECTIVE:

Student will be able

1. To understand the basic knowledge of opps with C++ language.
2. To understand the concept of data abstraction and encapsulation.
3. To learn how inheritance and virtual functions implement dynamic binding with polymorphism.
4. To understand the Structure & classes concepts, data member.
5. To understand the Array, Pointers operations.
6. To understand the Function overloading & Operator Overloading.
7. To understand the Inheritance & C++/I/O system.

### Syllabus:

#### UNIT - I

**Overview of C++** - Overview of C++, Software crisis, Object oriented programming paradigm, Basic concepts of OOP, Advantages/Benefits of OOP, and Usage/applications of OOP

**C++ Environment**, Program development environment, The language and the C++ language standards, Introduction to various C++ compilers, The C++ standard library, Prototype of main() function, i/o operator, manipulator, comments, data types

**Creating and Compiling C++ Programs** - TURBO C++ IDE, Creating, compiling and running a C++ program using ide and through command line, Elements of C++ Language, Structure of a C++ program, C++ tokens, Type conversion in expressions

#### UNIT - II

**Decision Making and Branching** - Introduction, Sequential statements, Mathematical Functions, Branching statements, looping Statements, Nested loops, Programming examples

**Arrays and Functions**- Arrays, The meaning of an array, Single-dimensional arrays, Two-dimensional arrays (Multi-dimensional arrays), User Defined Functions, Elements of user-defined functions, Return values and their types, Function calls, Categories of functions, Passing parameters to functions, Recursion, Command Line Arguments, Storage Class Specifies

#### UNIT - III

**Classes and Objects** - Classes, Structures and classes, Unions and classes, Friend function, Friend classes, Inline function, Scope resolution operator, Static class members, Static data members, Static member functions, Passing object to functions, Returning objects, Object assignment

**Array, Pointers, References and the Dynamic Allocation Operators** - Array of objects, Pointer to object, Type checking in C++, The this pointer, Pointer to Derived Types, Pointer to class members, References, C++'s Dynamic Allocation Operators

**Constructors and Destructors** - Introduction, Constructors, Default Constructor, Parameterized constructors, Copy Constructors, Multiple Constructors in a class, Constructors with default arguments, Default Arguments, Special Characteristics of Constructor functions, Destructors

**UNIT - IV**

**Function and Operator Overloading** - Function overloading, Overloading Constructor Function, Finding the address of an overloaded function, Operator Overloading, Creating a Member Operator Function, Creating Prefix and Postfix forms of the increment (++) and decrement (--) operators

(Overloading Unary Operator), Overloading the Shorthand Operators (i.e. +=, -= etc), Operator Overloading Restriction (Rules), Operator Overloading using friend function, Overloading new and delete operator, Overloading some special operators, Overloading [ ] (Subscripting) operator, Overloading() (Function Call) operator, Overloading Binary Arithmetic operators, Concatenating String, Overloading Comma (,) operator, Overloading the I/O operators

**UNIT - V**

**Inheritance** - Introduction to inheritance, Features or Advantages of Inheritance, Type of Inheritance :, Base Classes and Derived Classes, Base Class Access Control, Protected Members, Protected Base class Inheritance, Inheriting Multiple Base Classes, Constructors, Destructors and Inheritance, Passing Parameters to Base Class Constructors, Granting Access, Virtual Base Classes

**Polymorphism** - Polymorphism, Types of Polymorphism, Virtual Functions and Polymorphism, Pure Virtual Functions, Early Vs Late Binding

**The C++ I/O System Basics** - The C++ I/O System basics, C++ predefined streams, Formatting using the ios members, Clearing Format Flags, An Overloaded form of setf(), Examining the Formatted Flags, Using width(), Using precision(), Using fill(), Using Manipulators to format I/O, Creating your own Manipulators

**Practical's:**

1. WAP to add, subtract, multiply and divide two numbers using concepts of C++.
2. WAP to show swapping of two numbers using C++.
3. WAP to calculate volume of cube, cylinder, rectangular box using three times function overloading in C++.
4. WAP using virtual function.
5. WAP using copy constructor.
6. WAP to show multiple inheritances.
7. WAP to find mean value of two numbers using friend function.
8. WAP using inline function.
9. WAP to demonstrate the use of Local Object, Static Object & Global Object using C++.
10. WAP in C++ to demonstrate the creation and the use of dynamic object.
11. Derive the two classes son and daughter and, demonstrate polymorphism in action.

**COURSE OUTCOME:**

After Study This Student Will Be Able To Know About And Concepts of OOPs with C++ Language, Classes. Student will be able to create Arrays Its uses, Uses of function overloading, inheritance & C++I/O system.

**Text Books:**

- "Programming In C++", M Kumar TMH Publications
- "Object Oriented Programming With C++ ", R. Subburaj Vikas Publishing House, New Delhi.isbn 81-259-1450-1
- "Object Oriented Programming C++ " R. Lafore



**Reference Books:**

- “C++ The Complete Reference ” Herbert Schildt TMH Publication ISBN 0-07-463880-7
- “C++ ”, E. Balaguruswamy, TMH Publication ISBN 0-07-462038-x
- “Object Oriented Programming with ANSI & Turbo C++”, Ashok. N. Kamthane, Pearson Education Publication, ISBN 81-7808-772-3

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## DBMS & SQL

### COURSE OBJECTIVE:

Student will be able

1. To understand the basic knowledge of DBMS Concepts.
2. To understand the Database Design.
3. To understand the RELATIONAL DATA MODEL.
4. To understand the RELATIONAL DATABASE DESIGN.
5. To understand the Indexing & Hashing-Basic Concepts & Recovery System.

### Syllabus:

#### UNIT - I

**Introduction to DBMS & RDBMS** - Introduction to database, Introduction DBMS, Different database models, Structure of DBMS, RDBMS an introduction, Cod's law for RDBMS, Components of rdbms (kernel/data dictionary)

**Introduction to Oracle RDBMS and Client/Server Computing** - Introduction to Oracle, The Features of Oracle 9i, The oracle product details, An introduction to client/server computing, Oracle and client/server computing

**Overview of Oracle Architecture** - Oracle Architecture, Oracle Files, System and User Processes, Oracle Memory, System Database Object, Protecting Data

#### UNIT - II

**Introduction to SQL\*PLUS** -Introduction to SQL, Features of SQL, Components of SQL, Introduction to SQL\*PLUS, Features of SQL\*PLUS, Execution of SQL\*PLUS, Important commands used in SQL\*PLUS, Oracle Data-Types

**Working with Tables** -Tables - An Introduction, Use Of Table In SQL, Viewing The Stored Data In Tables, Filtering Table Data, Updating Data, Deleting Data From Tables, Modifying The Structure Of Tables, Destroying A Table, A Few Other SQL Statements

#### UNIT - III

**Data Constraints** - Data Constraints, The Use of Data Constraints, The Types of Data Constraints, Defining Integrity Constraints By 'Alter Table', Removing Integrity Constraints, 'Null' Value Concept, 'Not Null' Constraint, Default Value Concept, 'User Constraints' Table

**Data Manipulation in SQL** - Oracle Operators, Range Searching, Pattern Matching, LIKE 'IN' and 'NOT IN' Predicates, An Introduction to 'DUAL' Table, An Introduction to 'SYSDATE'

**Oracle Functions** - Oracle Function, Function Types, Group Function, Scalar Function, Working With 'Date' in SQL, Grouping Of Data Of Different Tables In SQL

**Joins, Sub-Queries & Views** - types of joins, use of sub-query, 'union' and clause, 'Intersect' Clause, Minus Clause, Concept of View, Types of View, Use of View

#### UNIT - IV

**User Accounts Management & Indexing** - Creation of User Account, User Account Management, Granting Privileges, Revoking Privileges, Modifying Password, Closing User Account, Concept of Index, Creation of Index, Types of Index, Use of Index, Deleting Index

## COURSE CODE: 6IMIT202

**Introduction to PL/SQL Programming** - Introduction to PL/SQL, Advantages of PL/SQL, Differences between SQL and PL/SQL, PL/SQL Block Structure, PL/SQL Character set, Variable,

Constant and Data type, Assignment Operator and the use of 'SELECT...INTO, PL/SQL Program Control Structure, The use of 'IF...THEN...ELSE...ENDIF', Iteration Control (The use of LOOP, WHILE, FOR), The use of 'GOTO Statement

**Cursor** - Cursor an Introduction, Types of Cursor, Features of Cursor, Implicit Cursor, Explicit Cursor, Application of for Loop with Cursor

## UNIT – V

**Exception Handling in PL/SQL** - Exception Handling in PL/SQL, Built in Exception Handling, User Defined Exception Handling, The Raise Application-error Procedure

**Oracle Transaction** - Oracle Transaction, Commit Statement, Rollback Statement, Save point statement, Concept of lock, Types of locks, Levels of Locks, 'SELECT.....FOR UPDATE' Statement, Removing the Lock

**Procedures and Functions-** Concept of Procedures and Functions, Advantages of Procedure and Function, Creation of Procedure and Function, Deleting Procedure and Function

**Database Triggers** - Concept of Triggers, Types of Triggers, Creation of Triggers, Application of Triggers, Deleting Triggers

## Practicals:

1. Write a query to implement Different types of DDL statements in SQL.
2. Write a query to implement Different types of DML statements in SQL.
3. Write a query to implement Different types of DQL statements in SQL.
4. Write a query to implement Different types of DCL statements in SQL.
5. Write a query to explore 'select' clause using where, order by, between, like, group-by, having etc.
6. Write a query to implement the concept of Joins in SQL.
7. Write a query to implement the concept of Indexes and views.
8. Write a query to implement the restrictions on the table.
9. Write a query to implement the concept of Sub Questionries.
10. Write a query to implement the structure of the table.

## COURSE OUTCOME:

After study this student will be able to know about and concepts & fundamentals of DBMS, Concept of keys, RELATIONAL DATA MODEL& design. **Student will also able to create table and implement commands.**

## Text Books:

- Database Systems : Concepts, Designs and Application Shio Kumar Singh
- Introduction to Database Management Systems Atul Kahate

## Reference Books:

- "SQL, PL/SQL", Ivan Bayross Bpb Publications"
- "The Oracle Cook Book", Liebschuty BPB Publication
- "Oracle A Beginners Guide". Michael Abbey, Michael J.Corey, TMH Publication
- Oracle Unleashed (Chapter 1, 2,3,4,5 and 9)

# DATA STRUCTURE

## COURSE OBJECTIVE:

Student will be able

1. To understand the basic knowledge of data structure.
2. To understand the Abstract data type concepts.
3. To understand the Linked List & its Basic operations.
4. To understand the Basic Terminology of TREES.
5. To understand the Analysis of algorithm & Introduction to graphs.

## Syllabus:

### UNIT - I

**Analysis of Algorithm**-Introduction, Criteria of Algorithm, Time Complexity, Space Complexity, Asymptotic Notation: Big Oh (O) Notation: Big Omega ( $\Omega$ ) Notation: Big Theta ( $\Theta$ ) Notation

**Types of Data structures**- Introduction, Types of Data structures, Linear Data Structures, Non Linear Data Structure, Array, SPARSE MATRICES, Garbage Collection, Benefits, Disadvantages

### UNIT – II

**Stacks**-Introduction, Push operation, Pop operation, Stack implementation using arrays, (static implementation of stacks), STACK as a Linked List, Stack as an abstract data structure, Applications of stack, Conversion of Expressions, Precedence and associativity of the operators, Evaluation of Postfix expression, Multiple stacks,

**Recursion**-Introduction, Working of recursion, Fibonacci series, Tower of Hanoi, Efficiency of recursion

**Queue**-Introduction, Different types of queues, Queue (Linear queue), Queue as an abstract data structure, Circular queue, Double ended queue (Deque), Priority queue, QUEUE as a Linked List, Applications of Queue

**Linked Lists**-Concept of list and array , Introduction to Data Structures, Arrays, Linked list, Singly or Linear linked list, Circular singly linked list, Doubly linked lists, Header Node, Applications of linked lists, Addition of two long positive numbers, Evaluation of a polynomial

### UNIT – III

**Trees**-Introduction, Representation of tree, Binary Tree , Representation of binary tree, Array representation of binary tree, Linked List representation of binary tree, Basic Operation on Binary Tree- Traversals, Binary Tree Traversal Algorithms (Recursive), Creation of Binary Search Tree:, Types of binary trees, Operations on Binary Search Tree (BST), Threaded binary trees, Application of Binary Tree:, B-Tree, Height Balanced Tree ,

**Graph**- Introduction to Graphs, Undirected Graph, Directed Graph or digraph, Graph Representation, Adjacency Matrix Representation, Adjacency List Representation, Graph Traversals, Breadth First Traversal, Depth First Traversal , Searching in Graph, Minimal Spanning Tree, Kruskal's Algorithm, Prim's Algorithm , Shortest Path in Graph,

### UNIT – IV

**Sorting and Searching** - Introduction, Bubble sort, Selection Sort, Merge Sort, Quick sort, Insertion Sort, Shell sort, Address calculation sort, Radix sort, Comparison of sorting methods, Hash Table,

## **COURSE CODE: 6IMIT203**

Collision Resolution Techniques, Linear Search (Sequential Search), Binary Search, Searching an ordered table, Indexed sequential search, Interpolation search

### **UNIT – V**

**File Structure And Indexing-** Introduction, Objectives, Terminology, File Organization, Sequential Files, Disadvantages, Direct File Organization, Indexed Sequential File Organization

### **COURSE OUTCOME:**

After study this student will be able to know about the concepts of Data Structure Using C++ Language, List & Its Operations Concept Of Tree, Algorithm & Graphs Design.

### **Text Books:**

- Weiss- Data Structures & Algorithm Analysis in C++ (A.W.L.)
- Data Structures, algorithms and Applications in C++ by Sahni (McGraw Hill)

### **Reference Books:**

- Fundamentals Of Data Structure, By S. Sawhney & E. Horowitz
- Data Structure : By Trembley & Sorrenson
- Data Structure : By lipschuists (Schaum's Outline Series McGraw Hill Publication)
- Fundamentals Of Computer Algorithm: By Ellis Horowitz and Sartaj Sawhney

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

# OPERATING SYSTEMS

## COURSE OBJECTIVE:

Student will be able

1. To develop the understanding of functioning of Operating System.
2. To understand the Process Concepts, process state & process control
3. To understand the Critical Section Problem
4. To understand the Contiguous Allocating, Paging
5. To understand the Disk Scheduling, Disk Management

## Syllabus:

### UNIT - I

**Operating Systems:** Overview -Introduction of Operating System, Types of Operating System, System Components and its services, System Calls, System Programs, Structure, Design and, Implementation, Operating System Generation

### UNIT - II

**Process :** Concept, Description and Control -Concept of process, Process state model, Process description - PCB, Process control, Threads, Threads in Linux

**Process Scheduling** - Types of Scheduler, Scheduling Criteria, Uniprocessor, Scheduling, Multiprocessor Scheduling, Algorithm Evaluation, Process Scheduling in Linux

**Concurrency** - Introduction to concurrency, Critical section problem, Mutual Exclusion solutions, S/w approach, H/w support, semaphore, monitor, Classical problem of synchronization

### UNIT – III

**Deadlock** - Deadlock Characterization, Deadlock Prevention, Deadlock Detection, Deadlock Avoidance, Combined Approach

**Protection** - Goal of Protection, Protection Domains, Access Matrix, Implementation of Access Matrix, Revocation of Access Rights, Language Based Protection

**Security and Encryption** - Security Problem, User Authentication, Program Threats, System Threats, Securing System and Facilities, Encryption & Decryption – Cryptography

### UNIT – IV

**Memory Management** -Memory Management Requirements, Address Space, Linking and Loading, Swapping, Partitioning, Paging, Segmentation

**Virtual Memory** - Introduction to Virtual Memory, Demand Paging, Page Replacement, Thrashing, Demand Segmentation

**Input Output Systems** - Input - Output Devices, Hardware Support for I/o, I/O Communication Techniques, I/O Software Device Drivers, Performance Consideration,

**Disk Structure** - Introduction to Disks, Disk Scheduling, Disk Management, Disk Reliability, Swap Space Management, Stable Storage Implementation

### UNIT – V

**File Management** - File Concepts, Directory structure, File Sharing, Protection, File system in Linux

**COURSE OUTCOME:**

After Study this student will be able to know about functioning of Operating System. To make students able to learn different types of operating systems along with concept of file systems and CPU scheduling algorithms used in operating system.

To provide students knowledge of memory management and deadlock handling algorithms. At the end of the course, students will be able to implement various algorithms required for management, scheduling, allocation and communication used in operating system.

**Text Books:**

- Operating System: Gary Nutt
- William Stallings
- Charles Crowley
- Pearson Education
- Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne

**Reference Books:**

- Operating System Concepts by Silberschatz & Galvin, Addison Wesley Publication 6th Edition.
- Operating System Concepts & Design by Milan Milen Kovic, TMH Publication

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## Communication Skill & Personality Development

**Objective:** To make the students understand the basics of personality, public speaking, language, Listening, conversation & writing skills, along with the communication process Syllabus

**THEORY –**

**Unit- I:**

Basics of Personality, Do's and Dont's in Personality, Salutations and Greetings, Presenting Yourself, Proper Introduction of Oneself.

**Unit- II:**

Administration- your work style, Overcoming Phobias, Public Speaking, General Etiquettes and Mannerism, Time Management, Attire, Attitude, Self Actualization, Magic of Positive Thinking.

**Unit- III :**

Tips of Preparing CV, Interviews tips.

**Unit-IV:**

Language Skill, Writing Skill, Speaking Skill, Listening Skill, Conversation Practice, Mysticism of Body Language, Basics of Grammar.

**Unit- V :**

Communication- Meaning, Functions, Channels, Process, Barriers and Interpersonal Skills.

**PRACTICAL –**

1. To present self introduction of yours.
2. Mock interview.
3. Group discussions.
4. SWOT analysis of self.
5. Extempore.
6. Debate.
7. Preparation of CV.
8. Role play.
9. Present a speech.
10. Make a power point presentation of communication.

**Reference Books:**

1. Business Communication, Universal Pub. Agra – Dr. Ramesh Mangal
2. English Grammar- Wren & Martin
3. Putting your best foot forward- Lt. Co. (Dr.) Pramod Deogirikar

---

**Chairperson**

**Dean (Academics)**

**(Board of studies)**

**(Academic Council)**

**(Registrar) Seal**

**Outcome-** After the completion of this subject the learners will understand the basics of personality, public speaking, language, Listening, conversation & writing skills, along with the communication process.



## PROGRAMMING WITH VB.NET

### COURSE OBJECTIVE:

Student will be able-

1. The aim of the course is for the student to gain knowledge in the basic concepts of object-oriented programming and build skills to develop modern software programs using the language Visual Basic. The course is also suitable for students with prior programming experience who wish to strengthen their knowledge in the area of object-oriented design and programming with Windows.
2. Analyze program requirements
3. Design/develop programs with GUI interfaces
4. Code programs and develop interface using Visual Basic .Net
5. Perform tests, resolve defects and revise existing code.

### Syllabus:

#### UNIT - I

Introduction to .NET - Introduction, What is a Program?, What is Programming?, What do you mean by .NET Framework?, Features of .NET Framework, VB 6 VS VB.NET, VB.NET VS JAVA, VB.NET VS C#, What is .NET Architecture?, What is CLR?, What do you mean by Class Library?, Versions of .NET Framework, What are Assemblies?, Namespaces, CTS (Common Type System), Interfaces, What is special in VB.NET?

Visual Studio **2010** - Introduction, What is Visual Studio?, Flavors of Visual Studio, Visual Studio **2010**, File Extensions Used in VB.Net, Using Visual Studio **2010**, Feature of Visual Studio **2010**, Output Window, Components Tray, References and the Reference Window, Quick View of Visual Studio **2010**, Opening an existing project, Adding a Form to a Project.

#### UNIT – II

The Visual Basic Language - Visual Basic Statements, Data Types in VB.NET, Declaring Variables, Declaration of Variables (Advanced), Data Type Conversion, String Functions, Formatting Data, Arithmetic Operators, Parentheses and Precedence, Operator Operation, Constants, Control Statements, Arrays in VB.NET, Specifying Optional Procedure Arguments, Passing a Variable Number of Arguments, Recursion, Using a Delegate

Working With The Controls - The Toolbox, Adding and deleting Tools in the Toolbox, creating a tab on the toolbox, Form Designer Basics, The Button, The Combo Box , The List Box, The Checkbox, The Picture Box, The Radio Buttons, The Scroll Bar, Timer, List View, Tree View, Toolbar, Dialog Boxes, Menus in VB.NET, Link Label Control

Designing Menus - Menus, Context menu, Event of the Menu Item, Creating menu items in Visual Studio .Net

#### UNIT – III

Object Oriented Programming with VB.NET - OOPs?, What is an Object?, What are Classes?, Visual Basic .NET and Object-Oriented, Principles of Object-Oriented Programming, Classes V/s Objects, Inheritance, Polymorphism and Overloading, Scope and Accessibility in Class Modules, Namespaces, Managed Execution, Assemblies, Assemblies in VB .NET

The .NET Framework Class Library - The .NET Framework Class Library, The System Namespace, Data Type Conversion Using Convert Class, The Array Class, The Math Class, The String Class, Other Namespaces, System. Collections, System. Data, System.IO

**UNIT – IV**

OLE/COM/Win32 API - Object Linking and Embedding, History of OLE/COM, Component Object Model (COM), COM interoperability in .NET, Win32 API in .NET, COM Interoperability in .NET, Installation and Registration of Assembly, Microsoft Office solutions with Visual Studio .NET, Automation of Office from Visual Studio .NET, Creating and opening Microsoft Word document from VB.NET  
User Controls in VB.NET - Introduction, The Control Class, The Control Class' Properties, The Control Class' Methods, Creating the Control Project 1, The RoundButton Control, Creating the Control Project 2, Building the new Button

**UNIT – V**

A Brief Introduction to Database Access with VB .NET - Introduction, What is ADO?, What is ADO.NET?, The Connection Object, Connecting to a Database, The Command Object, The DataAdapter Object, The DataReader Object, The DataSet Object, Updating Your Database by Using DataSets, The AcceptChanges () Method, The RejectChanges () Method, The HasChanges () Method, The GetChanges () Method, Working with DataSets in Visual Studio, Moving Around in DataSet and Retrieving Data, Using Strongly Typed DataSets, DataSets With Multiple Tables, Finding and Sorting Data in DataSets, Filtering on Row State and Version, Data View Manager.

**Practicals:**

1. Create an application to accept a character from console and check the case of the character
2. Write a VB.Net program to accept any character from keyboard and display whether it is vowel or not.
3. Write a VB .NET program to accept a string and convert the case of the characters
4. Develop a menu based VB .NET application to implement a text editor with cut, copy, paste, save and close operations.
5. Write a program to implement the calculator with memory and recall operations.
6. Develop a form in VB .NET to pick a date from calendar control and display the day, month, year in separate textboxes.
7. Develop a VB .NET application using the File and Directory controls to implement a common dialog box.
8. Develop a Database application to store the details of students using ADO.NET.
9. Develop a Database application using ADO.NET to insert, modify, update and delete operations.
10. Develop a VB.NET application using Datagrid to display record.

**COURSE OUTCOME:**

After the completion of the course, students are expected to:

1. Have gained a good understanding of the basic concepts of object orientation
2. Have a good understanding of the visual basic language structure and language syntax
3. Have developed the ability to design and develop interactive applications using the object-oriented principals, encapsulation, inheritance and to some extents polymorphism
4. Be able to effectively develop applications with full functionality and a graphical user interface using the language visual basic
5. Have the capability of analysing and finding suitable and effective solutions to windows based applications using classes and objects

**Text Books:**

- VB.NET Programming Hemant Bairagee
- Test Your VB.Net Skills Y.P. Kanetkar & A. Dani

**Reference Books:**

- VB.NET Programming Black Book Steven holzner dreamtech publications
- Mastering VB.NET Evangelos petroustos BPB publications
- Introduction to .NET framework Worx publication

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

# JAVA PROGRAMMING

## COURSE OBJECTIVE:

Student will be able

- To understanding the JAVA environment.
- To understand the basic concept of oops with java techniques.
- To understand the Classes and its parts of programming
- To understand the Applets and other concept of java.

## Syllabus:

### UNIT - I

**OVERVIEW OF JAVA** -Introduction, Programming paradigm, OOPS Concepts, Evolution of Java, Features of Java, C++ Vs Java, Java and Internet, Java and WWW, Java support systems, Java Environment

**KEY FEATURES OF JAVA** - Introduction, Java Program Structure, Simple Java Program, Tokens, Java Statements, Java Virtual Machine, Constants and Variables, Declaration of Variables, Scope of Variables, Data types, Symbolic Constants, Type Casting, Command line arguments

### UNIT - II

**OPERATORS** - Operators, Arithmetic Operators, Relational Operators, Logical Operators, Bitwise Operators, Increment and Decrement, Conditional Operators, Special Operators, Assignment Operators, Expression & its evaluation

**CONTROL STATEMENTS** - Introduction, Control Statements, Sequence Control Statement, Decision Control Statement, Case Control Statement, Iteration Control Statement, Jump in loops, Labelled Loops

**ARRAYS AND STRINGS** - Introduction, Array, Need of Array, Types of Array, One dimensional Array, Two-Dimensional Array, Multidimensional Array, Strings, Concatenation of Strings, Methods for String Comparison, Methods for searching Strings, Changing the case of characters, String Buffer

### UNIT - III

**CLASSES** -Introduction, Defining a Class, Adding Variables, Adding Methods, Creating Objects, Accessing Class members, Call by value and call by reference, Recursion, Access Control, Constructors, Method overloading, Constructor Overloading, Garbage Collection, finalize( ) method, this keyword, Static Members, Nesting of Methods

**INHERITANCE** - Inheritance, Single Inheritance, Multilevel Inheritance, Multiple Inheritance, Hierarchical Inheritance, Using Super, Constructor -Order of Execution in Inheritance, Overriding methods, Final variables and methods, Final Classes, Abstract methods and Classes, Containership, Visibility Control

**WRAPPER CLASSES AND VECTORS** - Introduction, Wrapper Classes, Number Class, Byte class, Short class, Integer class, Long class, Converting Numbers to and from Strings, Float class, Double class, Character class, Boolean class, Vectors, Creating a vector

### UNIT - IV

**INTERFACE & PACKAGES** - Introduction, Interfaces, Defining interface, Implementing interface, Accessing interface method, Accessing interface variable, Extending interfaces, Packages, System packages, Using system packages, User defined packages, Adding class to a package, Accessing and using package

**EXCEPTION HANDLING** - Introduction, Exceptions, Using try & catch, multiple catch clauses, Finally, Throw, Throws

**MULTITHREADING** - Introduction, The Main Thread, Creating Threads, Life cycle of Thread, Using Threads Methods, Thread Priorities, Stopping and Blocking a thread, Thread Exceptions, Using is Alive() and join(), Synchronization

### **UNIT-V**

**APPLETS** - Introduction, Local & remote applets, Applet vs applications, Writing applets, Life cycle of an applet, Creating source code of applet, Creating an executable applet, Creating applet tag, Adding applet tag to html, Running the applet, Detailed form of applet tag, Passing parameters to applet, Aligning the display, Html tags, Getting input from user

**INPUT-OUTPUT STREAMS AND FILE MANAGEMENT** - Introduction, Stream, Stream Classes, Byte Stream Classes, Character Stream Classes, System Class, Reading Console Input, Writing Console Output, Using the File Class, Random Access File

**GRAPHICS PROGRAMMING** - Introduction, The Graphics Class, Drawing Lines and Rectangles, Using drawOval() and fillOval() method, Drawing arcs, Drawing Polygon, Line Graphs, Drawing Bar Chart

### **Practicals:**

1. Write a Java Program to Display message on computer screen.
2. Write a Java Program to develop a class for Rational numbers
3. Design a Date class in Java
4. Write a Java Program to design an interface for Stack ADT and implement Stack ADT using both Array and Linked List.
5. To develop a vehicle class hierarchy in Java to demonstrate the concept of polymorphism
6. Design a Date class in Java.
7. To write a Java Program to randomly generate objects and write them into a file using concept of Object Serialization
8. Develop a scientific calculator using even-driven programming paradigm of Java.
9. To write a multi-threaded Java program to print all numbers below 100,000 that are both prime and Fibonacci number
10. To develop a Java Program that supports multithreaded echo server and a GUI client.
11. To implement a calculator using GUI Environment with the help of javax.swing package.

### **COURSE OUTCOME:**

Students will complete software projects comprised of an object-oriented design, implementation, and test plan.

- Designs will demonstrate the use of good object-oriented design principles including encapsulation and information hiding.
- The implementation will demonstrate the use of a variety of basic control structures including selection and repetition; classes and objects in a tiered architecture (user interface, controller, and application logic layers); primitive and reference data types including composition; basic AWT components; file-based I/O; and one-dimensional arrays.
- Test plans will include test cases demonstrating both black box and glass box testing strategies.

**Text Books:**

- Java Programming by Kamal Prakashan

**Reference Books:**

- Programming Java 2nd Edition E. balagurusvamy, TMH Publications.
- Peter Norton Quid E To Java Programming Peter Norton, Techmedia Publications.
- "Java How to program", Deitel&Deitel, Prentice Hall, 4 th Edition, 2000.
- "Core Java Vol 1 and Vol 2", Gary Cornell and Cay S. Horstmann, Sun Microsystems Press, 1999.
- "Developing Java Enterprise Applications", Stephen Asbury, Scott R. Weiner, Wiley, 1998.

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## A. SOFTWARE ENGINEERING

### COURSE OBJECTIVE: –

Student will be able-

- a. To understanding the concepts and methods required for the construction of large software intensive systems.
- b. To develop a broad understanding of the discipline of software engineering.
- c. To understanding the detailed knowledge of techniques for the analysis and design of complex software intensive systems.
- d. To understanding the techniques in an appropriate engineering and management context.
- e. To understanding the brief account of associated professional and legal issues.

### Syllabus:

#### UNIT - I

**Introduction to software engineering** - introduction, reusable software components, what is well engineered software? Programming and software engineering, what is software engineering?, goals of software engineering, software processes, software process models, process iteration, other important software models

**Software project management** - project management, management activities, project planning, project scheduling, risk management, selecting staff, metrics used for measuring the software cost, cocomo model

**Software process and project metric** - software quality, metrics for the analysis model, metrics for the design model, metrics for source code, metrics for testing

#### UNIT - II

**Software project planning** - introduction, software project planning, other planning activities, organization of the software project, management plan (smpm) document

**Software cost estimation** - introduction, software cost factors, programmer's ability, product complexity, product size, required level of reliability, level of technology, decomposition technique, empirical estimation models, the structure of estimation models

**Software project requirements** - software requirements, functional and non-functional requirements, user requirements, system requirements, software requirements document

**Requirements engineering process** - requirements engineering process, feasibility study, requirements elicitation and analysis, scenarios, requirements specification, ethnography, requirements validation, requirements management

#### UNIT - III

**Software prototyping** - software prototyping, prototyping in the software process, rapid prototyping techniques, user interface prototyping

**Analysis concept and modeling** - analysis modeling, context model, data modeling concepts, cardinality and modality, flow oriented diagram, data dictionary

**Design concepts and principles** - introduction, design within the context of software engineering, design process and design quality, design concepts, information hiding, functional independence, design classes, the design model, software patterns

**UNIT - IV**

**Software architecture** - software architecture data design, architectural styles and patterns, analyzing alternative architectural designs, mapping the requirements into a software architecture, architectural design

**Designing the user interface** - user interface, input design, end-user considerations for input design, output design, design principles, screens, forms, menu, messages, importance of code, data codification schemes, designing code less systems

Software quality management, software quality management, role of a software quality manager, ISO quality model, quality assurance standards, quality planning, quality control, software reviews, software reliability

**Verification and validation** - verification and validation, software testing, verification and validation planning, software inspections, automated static analysis, cleanroom software development

**UNIT - V**

**Software testing models** - software testing fundamentals, black-box and white-box testing, white-box testing, basis path testing, control structure testing, black-box testing, object-oriented testing methods

**Software testing strategies** - the strategic approach, the software testing strategy, strategic issues, unit testing, integration testing, validation testing, system testing, test automation

**Computer aided software engineering (CASE)** - computer aided software engineering (CASE), case workbenches, integrating case environment, need of software reuse: types of reuse, reuse

**COURSE OUTCOME:**

After study this student will be able to

- a. Carry out an evaluation and selection of projects against strategic, technical and economic criteria and use a variety of cost benefit evaluation techniques for choosing among competing project proposals. Approach project planning in an organized step by step manner and select an appropriate process model produce an activity plan for a project.
- b. Identify project risks, monitor and track project deadlines and produce a work plan and resource schedule.
- c. Plan the evaluation of a proposal or a product and manage people in software environments. Understand the importance of teamwork and quality management in software project management. Apply these project management tools and techniques in a diversity of fields such as new product and process development, construction, information technology, health care, and applied research.

**Text Books:**

- Software Engineering PankajJalote
- “Software Engineering: A Practitioner’s Approach” Pressma
- “Software Engineering” K KAggarwal

**Reference Books:**

- “Fundamentals of Software Engineering” by Mall B
- Software Testing: Principles and Practice Gopaldaswamy and Srinivasan, 817758121x. Publisher, Pearson Education India. ISBN, 817758121x.
- Software Testing Tools: Covering WinRunner, Silk Test, LoadRunner, JMeter and TestDirector with case Dr. K.V.K.K. Prasad, ISBN: 8177225324, Wiley Dreamtech, List Price: Rs. 279.00
- Basics of Software Project Management – NIIT,, Prentice Hall of India, ISBN 81-203-2490-0
- Software Project Management Bob Hughes & mike Cotterell, Tata McGraw Hill , ISBN – 0-07-061985-9



## B. COMPUTER ARCHITECTURE

### COURSE OBJECTIVE:

Student will be able

1. To understand the lower level abstraction of a computer system including digital logic, instruction set and assembly language programming.
2. To understand data representation, logic gates, simplification of logical expressions, design and analysis of simple combinational circuit such as decoders and multiplexers.
3. To understand the working of flip-flops and registers, design and analysis of simple synchronous sequential circuit, random-access and read-only memories, instruction set architecture and programming in assembly language.

### Syllabus:

#### UNIT - I

**INTRODUCTION**- Digital Computers, Computer Architecture, Computer Organization, Difference between Computer Architecture and Organization, Structure and Functions, Summary, Exercise  
**BASIC ARRANGEMENT OF A COMPUTER SYSTEM** - Computer, Basic organization of a computer system, Types of computers, Microprocessor ( $\mu$ p), Working of Microprocessor, Microprocessor 8085 Architecture, Speed of Microprocessors, Summary, Exercise

#### UNIT – II

**NUMBER SYSTEM AND ITS REPRESENTATION** -Data Types, Number System, Number System Conversion, Binary Arithmetic, Integer and Floating Point Representation Overflow Underflow  
**COMPUTER CODES** - Introduction, BCD (Binary Coded decimal) Code, EBCDIC Code, ASCII Code, Excess-3 Code, Gray Code, Error Detection Code, Error Correction Code  
**BASIC BUILDING BLOCKS** - Logic Gates , Universal Gates, Exclusive Gates, Bubbled Gates, Universality of NAND and NOR gates  
**BOOLEAN ALGEBRA** - Boolean Variable, Boolean Algebra, Boolean Functions and Truth Tables, Logic Diagram, Laws of Boolean Algebra, Rules for Boolean Algebra, DeMorgan's theorems, Simplification of Boolean Functions, Implementation Using Basic Gates, To Obtain Expression from Logic Circuits

#### UNIT – III

**KARNAUGH MAP** - Introduction, Algebraic Expression by Karnaugh Map, Simplification of Boolean Expression using K Map, Simplification of Boolean expression using K-map, Don't care conditions  
**DIGITAL LOGIC CIRCUITS** - Introduction, Combinational Logic Circuit, Sequential Logic Circuits  
**BASIC COMPUTER ORGANIZATION** - Register Transfer language and Micro-operations, Instruction Codes , Instruction Set, Operations and Operands, Computer Registers, Instruction Format, Instruction Cycle, Addressing Modes, Real and Protected Addressing Modes, Assembly Language Programming, Input-Output and Interrupt

**UNIT – IV**

**Von Neumann Architecture-** Data Path and Memory Bus, Arithmetic and Logic Unit (ALU), Memory, Static Random Access Memory (SRAM), Dynamic Random Access Memory (DRAM), Control Unit (CU), Register Transfer Language, Execution of Instructions, Micro architecture, Complex and reduced instruction sets (CISC/RISC), Input/output.

**CENTRAL PROCESSING UNIT DESIGN** - Central Processing Unit (CPU), BUS Organization, Register Organization, Stack Organization, Data Path and Control Signals, Types of Processor (CPU) , Micro Programmed Control and Hardwired Control, Pipelining, Software - Hardware Interaction layers in Computer Architecture.

**INPUT-OUTPUT ORGANIZATION**, Transfer of Information between I/O Devices, CPU & Memory, Data Transfer Format , Types of Data Transfer , I/O Interface , Modes of Data Transfer , I/O Channels and Processors, Input/output Identification (Peripheral or Memory Mapped), Conditions of Data Transfer.

**UNIT – V**

**MEMORY ORGANIZATION** - Computer Memory, Characteristics of Memory, Units of Memory, Data Accessing/Storing Methods in Computer Memory, Memory Hierarchy, Classification of Memory, Associative Memory, Virtual Memory, Memory Management System

**Optimizing Hardware Performance** - Memory Hierarchy, Cache, Virtual Memory, Pipelining, Pipelining Hazards, Conclusion, Superscalar CPU, Brief Historical Detour into Supercomputing, Superscalar Principle.

**COURSE OUTCOME:**

After study this student will be able to know about the basic computer organization, design and micro-operations, Understanding of CPU functioning and computer arithmetic. Learning various methods and techniques of memory organization.

**Text Books:**

- Computer System Design & Architecture- Heuring Jordan(A.W.L.)
- Computer System Architecture- M.Morris Mano, PH.I.

**Reference Books:**

- Computer Organization &Architecture - William Stallings.
- Intel Microprocessors - Architecture, Programming & Interfacing-Barry. b Brey.
- Computer Organization                      ISRD Group                      Tata Mc Graw Hill

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## **A. COMPILER DESIGN**

### **COURSE OBJECTIVE:**

Student will be able-

- a. To introduce the major concept areas of language translation and compiler design.
- b. To enrich the knowledge in various phases of compiler and its use, code optimization techniques, machine code generation, and use of symbol table.
- c. To extend the knowledge of parser by parsing LL parser and LR parser.
- d. To provide practical programming skills necessary for constructing a compiler.

### **Syllabus:**

#### **UNIT - I: LEXICAL ANALYSIS**

Introduction to Compiling- Compilers-Analysis of the source program-The phases-Cousins-The grouping of phases-Compiler construction tools.The role of the lexical analyzer- Input buffering-Specification of tokens-Recognition of tokens-A language for specifying lexical analyzer.

#### **UNIT - II: SYNTAX ANALYSIS AND RUN-TIME ENVIRONMENTS**

Syntax Analysis- The role of the parser-Context-free grammars-Writing a grammar-Top down parsing-Bottom-up Parsing-LR parsers-Constructing and SLR (1) parsing table.  
Type Checking- Type Systems-Specification of a simple type checker.Run-Time Environments-Source language issues-Storage organization-Storage-allocation strategies.

#### **UNIT - III: INTERMEDIATE CODE GENERATION**

Intermediate languages-Declarations-Assignment statements - Boolean expressions-Case statements- Back patching-Procedure calls

#### **UNIT - IV: CODE GENERATION**

Issues in the design of a code generator- The target machine-Run-time storage management-Basic blocks and flow graphs- Next-use information-A simple code generator-Register allocation and assignment-The dag representation of basic blocks -Generating code from dags.

#### **UNIT - V: CODE OPTIMIZATION**

Introduction-The principle sources of optimization-Peepphole optimization- Optimization of basic blocks-Loops in flow graphs- Introduction to global data-flow analysis-Code improving transformations.

### **COURSE OUTCOME:**

- a. To apply the knowledge of lex tool &yacc tool to develop a scanner & parser.
- b. To design & conduct experiments for Intermediate Code Generation in compiler.
- c. To design & implement a software system for backend of the compiler.

- d. To deal with different translators.
- e. To develop program to solve complex problems in compiler.
- f. To learn the new code optimization techniques to improve the performance of a program in terms of speed & space.
- g. To acquire the knowledge of modern compiler & its features.
- h. To learn & use the new tools and technologies used for designing a compiler.

**Text Books:**

- “Crafting a Compiler with C”, C. N. Fisher and R. J. LeBlanc Pearson Education, 2000.
- “Modern Compiler Design”, David Galles, Pearson Education Asia, 2007
- “Advanced Compiler Design & Implementation”, Steven S. Muchnick, Morgan Kaufmann Publishers, 2000.

**Reference Books:**

- Introduction To Automata Theory, Language And Comutation - John E - Hopcroft, Rajeev Motwani, Jeffery D. Ullman 2nd Edition
- Compiler Construction Principles & Practice – D.M. Dhamdhare 2nd Edition
- Principles Of Compiler Design – Affred V. Aho, Jeffery D. Ullmancompilers Principles, Techniques And Tools – Affred V. Aho Ravi Sethi, Jeffery D. Ullman
- “Compilers- Principles, Techniques, and Tools”, Alfred V. Aho, Ravi Sethi Jeffrey D. Ullman, Pearson Education Asia, 2007.

---

**Chairperson  
(Board of Studies)**

**Dean  
(Academic Council)**

**(Registrar)  
Seal**

## B. ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEM

### COURSE OBJECTIVE:

Student will be able-

- a. To understanding the importance of AI and puzzle problem.
- b. To understanding the Search Techniques.
- c. To understanding the Symbolic and Statistical Reasoning.
- d. To understanding the frames and Structural Knowledge Representation.
- e. To understanding the expert system life cycle.

### Syllabus

#### UNIT - I

**Introduction to Artificial Intelligence**, Overview and definition of AI, Importance of AI, Early work in AI, General issues in AI, Problems of AI, AI Techniques, Scope and Application areas of AI  
**Problem Solving and Search**, State Space Search for problem solving, Production System, Search and Control Strategies, Breadth First Search, Depth First Search, Heuristic Search, Production System Characteristics, Problem characteristics, Some other Control Strategies, Uniform cost search, Depth-limited search, Iterative deepening search, Adversarial Search, Two agent Games, The Minimax procedure, Example Problems, -Puzzle Problem and Playing Chess, Traveling Salesman Problem, Tic-Tac-Toc Problem, Water Jug Problem

#### UNIT – II

**Heuristic Search Techniques**, Introduction, A General Graph Searching Algorithm, Generate and Test, Hill Climbing Search, Best First Search and A\* algorithm, Admissibility of A\*, Monotone or Consistency Condition, Problem Reduction, AND-OR TREE, Constraint Satisfaction, Cryptarithmic Problem  
**Knowledge Representation and Logic**, Introduction and Importance of Knowledge, Characteristics of Knowledge, Explicit and Implicit Knowledge, Declarative or Procedural knowledge, Internal vs. External Knowledge, Mappings and Knowledge representation Methods, Issues in Knowledge representation, Important Attributes, Relationship among attributes, Granularity of representation, Representing set of objects, Finding the Suitable structure

#### UNIT – III

**First order logic or predicate calculus**, Introduction, Syntax and Semantix, Extentions and Notational Variations, Representing Simple facts in Predicate Logic, Representing Instance and ISA Relationships, Inference in First Order Logic, Inference Rules Involving Quantifiers, Generalized Modus Ponens, Canonical Form, Unification, Forward and Backward Chaining, A Complete Inference Procedure: Resolution, The Resolution Inference Rule, Canonical Form for Resolution, Resolution Proofs, Conversion to Normal Form (Skolimization), Conversion to Clause Form, Resolution control strategies, Unit Preference, Set of support, Linear Input Resolution, Subsumption, Natural Deduction and Question Answering  
**Knowledge Representation Using Rules**, Representing Knowledge Using Rules, Procedural V/S Declarative Knowledge, Logic Programming, Forward and Backward Reasoning, Matching, Indexing, Matching with Variable, Complex and Approximate Matching, Conflict Resolution, Control Knowledge

**Symbolic and Statistical Reasoning**, Symbolic Reasoning under Uncertainty, Introduction to Reasoning, Nonmonotonic Reasoning and its Logics, Implementation Issues, Implementation: Depth – First Search, Implementation: Breadth – First Search, Statistical Reasoning, Symbolic Verses Statistical Reasoning

#### **UNIT - IV**

**Structural Knowledge Representation**, Weak Slot and filter structures, Semantic nets, Intersection search, Non-binary predicates Representation, Essential distinctions, Partitioned semantic nets, Semantic nets to Frames, Frames, Frames as sets and instances, Additional ways of relating classes to each other, Slots and full-fledged objects, Property Inheritance algorithm, Languages for Frame, Strong slot and filter structures, Conceptual Dependency, Scripts, CYC

**NLP : Natural Language Processing**, Introduction, Computational linguistics, Problems of NLP, NLP Steps, Syntactic processing, Grammars, parsers, One or many Interpretations, Parsing techniques, Transition networks and augmented transition net, Unification Grammar, Semantic analysis, Semantic grammars, Case grammars, Conceptual parsing, Semantic interpretation, Discourse & pragmatic processing, Focus use in understanding, Modeling beliefs, Use of goals and plans for understanding, Acts of speech, Postulates of conversation

#### **UNIT – V**

**Expert system**, Introduction, Need and Justification, Benefits of using ES, Characteristics, Applications, Building blocks of Expert system, Knowledge Base, Inference Engine, User Interface, Expert System Life Cycle, Representing and Using Domain Knowledge, Knowledge Engineering and Acquisition, Expert System Tools, Expert System Shells, Case Study: Mycin&Dendral, Rule Based Systems, Learning Procedure

**PROLOG: AI Programming Language**, Introduction, Data Types & Structures: Atom, Variables, Lists, Prolog Syntax and Programming, Prolog Objects and Methods, Objects & Relationships using Trees and Lists, Facts, rules, Relationships and queries, ‘IS’ Operator & Singleton Variable, ‘CUT’ Operator

#### **COURSE OUTCOME:**

After study this student will be able to know about the AI with search algorithm and expert system with life cycle.

#### **Text Books:**

- Artificial Intelligence, Rich E and Knight K, TMH, New Delhi
- Artificial Intelligence Elaine Rich and kevin Knight and Shivashankar B. Nair

#### **Reference Books:**

- Principles of Artificial Intelligence Nils J. Nilsson
- Introduction to Artificial Intelligence & Expert Systems Dan W. Patterson

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## C. DATA WAREHOUSING & MINING

### COURSE OBJECTIVE:

Students will be able

1. To understand the scope and necessity of Data Mining & Warehousing for the society.
2. To understand the designing of Data Warehousing so that it can be able to solve the root problems.
3. To understand various tools of Data Mining and their techniques to solve the real time problems.
4. To develop ability to design various algorithms based on data mining tools.

### Syllabus:

#### UNIT - I

**Strategic Information Management**, Need for strategic information, Decision support system, Knowledge discovery & decision making, Need for data warehouse, Definitions of Data warehousing and data mining, Common characteristics of Data warehouse, Data Marts, Metadata, Operational versus analytical databases, Trends and planning of Data warehousing.

#### UNIT - II

**Data Modeling Strategy**, Defining business requirements, Data modeling strategy, Fact tables, Dimensions, Star schema and other schemas, Multi dimensional data models, Data Cube presentation of fact tables, Using the Data warehouse, Designing tools for Data warehouse, OLAP models and operations.

#### UNIT - III

**Data Warehouse Architecture Components and Implementation Options**, Architectural components, Infrastructure: Operational & Physical, Extraction, Transformation and Loading, Components of an Oracle Data warehouse, Data Transformation Functions, DBA responsibilities, Capacity Planning. .

#### UNIT - IV

**Data Warehouse Implementation**, Implementation of Data warehouse, Physical design: steps, considerations, physical storage, indexing, Performance Optimization, Data warehouse deployment activities, Data security, Backup and recovery concepts, Data warehouse Maintenance.

#### UNIT - V

**Data Mining**, Basics of data mining, Related concepts, Data mining techniques, Data Mining Algorithms , Classification, Clustering and Association rules, Knowledge Discovery in databases( KDD) Process, Introduction to Web Mining

### COURSE OUTCOME:

After study this student will be able to know about the

- a. Process raw data to make it suitable for various data mining algorithms.
- b. Discover and measure interesting patterns from different kinds of databases.
- c. Apply the techniques of clustering, classification, association finding, feature selection and visualization to real world data.

**Text Books:**

- Data Mining Techniques, Arun.K.Pujari, University Press
- Data Mining Technique & Trend, N.P Gopalan, PHI
- Introduction to Data Mining, Tan, Pearson

**Reference Books:**

- Data Warehousing Fundamentals, by PaulrajPonnian, John Wiley.
- Data warehousing with oracle by Sima Yazdani – Shirley s. Wong
- Data Mining Concepts and Techniques, Han Kamber, Morgan Kaufmann
- Introduction to Business Intelligence and Data Warehousing, PHI
- The Data Warehouse Lifecycle toolkit, Ralph Kimball, John Wiley.
- 

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**



## **WEB DEVELOPMENT**

### **COURSE OBJECTIVE:**

1. To understand to develop web application using open source technologies
2. To understand XML scripting language and deploying application on Apache Web Server
3. To understand Web Server configuration
4. To understand MySQL database deployment for web applications.

### **Syllabus:**

#### **UNIT - I: Introduction and Web Development Strategies**

History of Web, Protocols governing Web, Creating Websites for individual and Corporate World, Cyber Laws, Web Applications, Writing Web Projects, Identification of Objects, Target Users, Web Team, Planning and Process Development.

#### **UNIT - II: HTML, XML and Scripting**

List, Tables, Images, Forms, Frames, CSS Document type definition, XML schemes, Object Models, Presenting XML, Using XML Processors: DOM and SAX, Introduction to Java Script, Object in Java Script, Dynamic HTML with Java Script.

#### **UNIT - III: Java Beans and Web Servers**

Introduction to Java Beans, Advantage, Properties, BDK, Introduction to EJB, Java Beans API  
Introduction to Servlets, Lifecycle, JSDK, Servlet API, Servlet Packages: HTTP package, Working with Http request and response, Security Issues.

#### **UNIT - IV**

JSP Introduction to JSP, JSP processing, JSP Application Design, Tomcat Server, Implicit JSP objects, Conditional Processing, Declaring variables and methods, Error Handling and Debugging, Sharing data between JSP pages- Sharing Session and Application Data.

#### **UNIT – V**

Database Connectivity, Database Programming using JDBC, Studying Javax.sql.\*package, accessing a database from a JSP page, Application-specific Database Action, Developing Java Beans in a JSP page, introduction to Struts framework.

**Practicals:**

1. Implements Basic HTML Tags
2. Implementation of Table Tag
3. Implementation of FRAMES
4. Design A FORM In HTML (Yahoo registration form)
5. Validation of FORM Using Java Script.
6. Program for exception handling using multiple catch statements and also create your Own exception.

## SOFT COMPUTING

### COURSE OBJECTIVE:

Students will be able  
Fundamental concepts used in Soft computing. The concepts of Fuzzy logic (FL) will be covered first, followed by Artificial Neural Networks (ANNs) and optimization techniques using Genetic Algorithm (GA). Applications of Soft Computing techniques to solve a number of real life problems will be covered to have hands on practices. In summary, this course will provide exposure to theory as well as practical systems and software used in soft computing.

### Syllabus:

#### UNIT – I

**Introduction-** What is soft computing, important soft computing techniques.

#### UNIT – II

**Artificial Neural Network** :Biological neural network Vs Artificial neural network, Neuron Model and Neural Network Architectures, ANN terminologies, ANN benefits, Supervised learning network :Error back propagation network, Perceptron learning (single layer only), Unsupervised learning network :Kohonenself organizing feature maps (SOM)

#### UNIT – III

**Fuzzy Logic**-Crisp set Vs Fuzzy set, Operations on Fuzzy sets, Fuzzy relation, Membership function, Fuzzy arithmetic and Fuzzy measures.

#### UNIT – IV

**Genetic Algorithm** – Introduction, representations of GA by binary and real-valued numbers, Genetic ,OPERators and Parameters: Selection, crossover, mutation, elitism, Genetic Algorithms in Problem Solving.

#### UNIT – V

**Swarm Intelligence:** Meaning, Particle Swarm Optimization: basics, terminology, problem solving usingPSO

### COURSE OUTCOME:

- Fuzzy logic and its applications.
- Artificial neural networks and its applications.
- Solving single-objective optimization problems using GAs.
- Solving multi-objective optimization problems using Evolutionary algorithms (MOEAs).
- Applications of Soft computing to solve problems in varieties of application domains.

**Text Books:**

- Soft Computing and Intelligent Systems: Theory & Applications, N.K. Sinha& M. M. Gupta(Eds), Academic Press, 2000.
- Principles of soft computing , S.N.Shivanandan and S.N. deepa Wiley India publication ,First Indian edition ,2008.

**Reference Books:**

- A Comprehensive Foundation to Neural Networks , Simon Haykins , Prentice Hall
- Fuzzy Sets and Fuzzy Logic: Theory and Applications , G. J. Klir, and B. Yuan, PHI learning ,2011.
- Fuzzy Logic and Fuzzy Decision Making: Concepts and Applications, Dr.G.Canon, Galgotia Publication.
- Genetic Algorithms in Search, Optimization, and Machine Learning, D. E. Goldberg, Addison-Wesley, 1989.
- :Neuro-Fuzzy and soft computing :A computational Approach to learning and machine intelligence , Jang,Sun and Mizutani PHI learning ,2011.

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## ADVANCED JAVA PROGRAMMING

### COURSE OBJECTIVE:

Student will be able

1. To understanding the JAVA environment.
2. To understand the Applets and applications.
3. To understand the Java packaging
4. To understand the Socket Programming
5. To understand the AWT.

### Syllabus:

#### UNIT - I

EXCEPTION HANDLING - Introduction, Exceptions, Using try & catch, Multiple catch clauses, Finally, Throw, Throws

MULTITHREADING - Introduction, The Main Thread, Creating Threads, Life cycle of Thread, Using Threads Methods, Thread Priorities, Stopping and Blocking a thread, Thread Exceptions, Using is Alive() and join(), Synchronization

#### UNIT - II

APPLETS - Introduction, Local & remote applets, Applet vs applications, Writing applets, Life cycle of an applet, Creating source code of applet, Creating an executable applet, Creating applet tag, Adding applet tag to html, Running the applet, Detailed form of applet tag, Passing parameters to applet, Aligning the display, Html tags, Getting input from user

#### UNIT - III

Using Standard Java Packages, Exploring java Input Output Classes

Exploring Java.util package- Interface summary, Class summary, Collection Framework, Core collection Interface, Set Interface, List Interface, Map Interface, Stored Map, and Interface Iterator.

Networking in Java-Network Basics, Java & Networking – Sockets & Ports, Client Server architecture, TCP, UDP, Server Sockets ad Datagram, Networking classes in JDK

#### UNIT - IV

TCP/IP & Datagram Programming in Java – Socket Programming Basics, Datagram Communication, TCP/IP Socket Programming in Java.

AWT – AWT Programming basics, working with Windows, Graphics & Text.

Multimedia in Java- Handling images, sound and animations in Java

#### UNIT - V

Event Handling in Java-Handling of various events in Java, Handling Mouse & Keyboards events Swing Classes.

### Practicals:

1. Write a Java Program to Display message on computer screen.
2. Write a Java Program to develop a class for Rational numbers

## **COURSE CODE: 6IMIT402**

3. Design a Date class in Java
4. Write a Java Program to design an interface for Stack ADT and implement Stack ADT using both Array and Linked List.
5. To develop a vehicle class hierarchy in Java to demonstrate the concept of polymorphism
6. Design a Date class in Java.
7. To write a Java Program to randomly generate objects and write them into a file using concept of Object Serialization
8. Develop a scientific calculator using even-driven programming paradigm of Java.
9. To write a multi-threaded Java program to print all numbers below 100,000 that are both prime and Fibonacci number
10. To develop a Java Program that supports multithreaded echo server and a GUI client.
11. To implement a calculator using GUI Environment with the help of javax.swing package.

### **COURSE OUTCOME:**

Students will complete software projects comprised of an object-oriented design, implementation, and test plan.

- A. Designs will demonstrate the use of good object-oriented design principles including encapsulation and information hiding.
- B. The implementation will demonstrate the use of a variety of basic control structures including selection and repetition; classes and objects in a tiered architecture (user interface, controller, and application logic layers); primitive and reference data types including composition; basic AWT components; file-based I/O; and one-dimensional arrays.
- C. Test plans will include test cases demonstrating both black box and glass box testing strategies.

### **Text Books:**

- Java Programming by Kamal Prakashan
- Programming Java 2nd Edition E. balagurusvamy, TMH Publications.

### **Reference Books:**

- Peter Norton Quid E To Java Programming Peter Norton, Techmedia Publications.
- "Java How to program", Deitel&Deitel, Prentice Hall, 4 th Edition, 2000.
- "Core Java Vol 1 and Vol 2", Gary Cornell and Cay S. Horstmann, Sun Microsystems Press, 1999.
- "Developing Java Enterprise Applications", Stephen Asbury, Scott R. Weiner, Wiley, 1998.
- Java : The Complete Reference : Comprehensive Coverage of the Java Language Herbert Schildt

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**

## NUMERICAL METHODS

### COURSE OBJECTIVE: –

Student will be able

The aim is to teach the student various topics in Numerical Analysis such as solutions of nonlinear equations in one variable, interpolation and approximation, numerical differentiation and integration, direct methods for solving linear systems, numerical solution of ordinary differential equations.

### Syllabus:

#### UNIT - I

**Algebraic Equation** :Computer Arithmetic – Floating point Numbers- Operations Normalization and their consequences. Iterative Methods – Roots of a Single transcendental equations and roots of Polynomials using Bisection Method , False position Method , Newton Raphson Method.

#### UNIT - II

**Simultaneous Algebraic Equation** :Gauss Elimination Method, Gauss-Jordan Method, Factorization Method, Jacobi's Iteration Method, Gauss- seidal Iteration Method.

**Matrix Inversion & Eigen Value**: Gauss Jordan Method, Factorization Method and Eigen Vectors.

#### UNIT - III

**Interpolations**: Polynomials interpolation, Newton Method. Lagrange's Interpolation Formula and difference tables.Least Square Approximations- Linear regression only.

#### UNIT - IV

**Differentiation and Integration**- Formula for Numerical Differentiation and Numerical integration by Trapezoidal Rule and Simpson's rule only.

#### UNIT - V

**Numerical Solution of Differential Equation** :- Euler's Method, Taylor series Method, RungeKutta Method.

### COURSE OUTCOME:

- 1) Knowledge and Understanding: Students are able to understand the nature and operations of Numerical Analysis, demonstrate familiarity with theories and concepts used in Numerical Analysis and identify the steps required to carry out a piece of research on a topic in Numerical Analysis, derivation of the Numerical Methods, studying their convergence rate and performance, applicability of the methods on different test examples.
- 2) Intellectual Skills: By the end of the course the student is expected to solve real-life and Engineering applications reflecting the student ability to:
  - Recognize and apply appropriate theories, principles and concepts relevant to Numerical Analysis.
  - B2 Critically assess and evaluate the literature within the field of Numerical Analysis.
  - B3 Analyze and interpret information from a variety of sources relevant to Numerical Analysis.

**COURSE CODE: 6IMMA401**

- 3) **Practical Skills:** By the end of the course student will have the ability to compare the computational methods for advantages and drawback, choose the suitable computational method among several existing methods , implement the computational methods using any of existing programming languages, test such methods and compare between them, identify the suitable computational technique for a specific type of problems and develop the computational method that is suitable for the underlying problem.
- 4) **Transferable Skills:** Within the lectures the student is able to transfer ideas and experience Numerical Analysis Techniques, work effectively both in a team and independently, apply the best computational methods to solve real-life and Engineering applications via computational packages such as MATLAB or Mathematica and develop his ability to self appraise and reflect on practice relevant to Numerical Analysis.

**Text Books:**

- Numerical Methods By S.S. Shastri, 4th edition, 2005, PHI publications.
- Numerical Methods in Engineering and Science, 36th Edition, Khanna Publishers, Delhi  
Computer Based Numerical and Statistical techniques, P.K.Mittal and Mukesh B., Galgotia Publication

**Reference Books:**

- Numerical Methods                      V. Rajaraman,                      3rd Edition, Prentice-Hall India Pvt. Ltd.

---

**Chairperson**  
**(Board of Studies)**

**Dean**  
**(Academic Council)**

**(Registrar)**  
**Seal**



**MAJOR PROJECT**

**COURSE OBJECTIVE:**

**Syllabus:**

**UNIT – I**

**UNIT – II**

**UNIT – III**

**UNIT – IV**

**UNIT - V**

**COURSE OUTCOME:**

---

**Chairperson  
(Board of Studies)**

**Dean  
(Academic Council)**

**(Registrar)  
Seal**