

INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES

DEVI AHILYA UNIVERSITY, INDORE

M. Tech.(IT) 5 Years

Semester -V

| Sub. Code | Sub. Name | L | T | P | C |
|------------------|--|----------|----------|----------|----------|
| IT-501C | Computer Architecture | 3 | 1 | 0 | 4 |
| IT-502A | Microprocessor and Assembly Language | 3 | 1 | 0 | 4 |
| IT-505B | Programming in Java | 3 | 1 | 0 | 4 |
| IT-511 | System Analysis and Design | 3 | 1 | 0 | 4 |
| IT-512 | Discrete Structures | 3 | 1 | 0 | 4 |
| IT-507C | Programming in Java Lab | 0 | 0 | 4 | 2 |
| IT-508E | Microprocessor and Assembly Language Lab | 0 | 0 | 4 | 2 |
| IT-509 | Comprehensive Viva | 0 | 0 | 0 | 4 |
| | | | | | 28 |

INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, DAVV, INDORE
M. Tech. (IT) 5 Years V SEMESTER
IT-501C Computer Architecture

Aim of Course: To understand the concepts of design and analysis of the hardware of a computer system and its components such as control unit, arithmetic and logical (ALU) unit, input/output, and memory unit.

Objectives:

The course is designed to make students:

- Learn concepts of microprogramming in the design of the central processing unit of a computer system.
- Understand various ways for interconnecting I/O devices to the system.
- Understand basic concepts of parallel processing

Course Contents:

UNIT I

Introduction and vocabulary, History of computer architecture, Overview of computer organization, Difference between Computer architecture & organization, von Neumann/Turing, IBM 360 series, Moore's law, Performance measurement: IPC, CPI, MIPS, Amdahl's law, CPU performance equation, Speeding it up, Performance Mismatch & Solutions, Instruction cycle, Interrupt cycle, Bus interconnections: Types, Arbitration, PCI.

UNIT II

CPU Structure, Registers, User Visible Registers, General Purpose Registers, accumulator organization, general register organization, stack organization of CPU, High level issues in CPU design, Memory: Location, Capacity, Unit of transfer, Access method, Performance (Access, cycle, transfer rate), Physical type (semi conductor or magnetic), Physical characteristics (volatile, erasable etc.), Locality of references, Cache mapping techniques, Cache write policies, Cache initialization, External memory, RAID organization of hard disks.

UNIT III

Input/Output: Programmed I/O, Interrupt Driven I/O, Direct Memory Access. Representing information digitally, Byte Ordering: Big-Endian & Little-Endian. Instruction sets, Elements of an Instruction, Instruction Representation, Instruction types, Number of Addresses, Design Decisions [CISC/RISC], Addressing Modes, Large Register File in RISC.

Register and data flow design, data fetch and instruction fetch in indirect instruction cycle, CPU control unit, Functions of Control Unit, Micro-Operations, Micro Programmed Control and Hardwired control unit and their advantages-disadvantages.

UNIT IV

Instruction level parallelism: Pipeline design, Synchronous & Asynchronous Pipeline conflicts: Resource conflict, Data dependency, and Branch difficulties. Solutions to deal with pipelining: Hardware interlocks, operand forwarding, Delayed load, Pre fetch target instruction, Branch target buffer, Loop buffer, Branch prediction, and Delayed branch. Super scalar design; Super pipelining, and VLIW processors.

UNIT V

Parallel Processing, Flynn's classification: SISD, SIMD, MISD, MIMD. Vector processor, Array Processor, Symmetric multi processing, NUMA, Cache coherence in parallel computing.

Reference Books:

1. William Stallings, Computer Organization and Architecture: Design for performance 8th Ed., Pearson Education.
2. Rajkamal, Computer Architecture, ISP 2006, Tata McGraw HILL.
3. Andrew Tanenbaum, Structured computer organization, 4th Ed., Prentice – Hall, Upper Saddle River, NJ, 2000. (Alternate reference)
4. M. Morris Mano, Computer System Architecture, 3rd Ed., Pearson Education.
5. Kai Hwang, Computer Architecture

INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, DAVV, INDORE
M. Tech. (IT) 5 Years V SEMESTER
IT-502A: Microprocessor & Assembly Language

Aim of Course: To introduce the basic concepts of microprocessor and assembly language programming.

Objectives:

The course is designed to make students:

- Develop an understanding of the operation of microprocessors.
- Learn assembly language programming.
- Learn the internal organization of some popular microprocessors.

Course Contents:

UNIT I

Microprocessor–Based Systems: Hardware and Interfacing, Microprocessors, Microcomputers and Assembly Language 8085, Architecture & Memory Interfacing I/O Devices.

UNIT II

Instruction Set and Addressing modes: Data transfer, Arithmetic, Logical, Branch & Machine control instructions, related programs & Addressing modes.

Additional Programming Techniques and Stack Operations: Subroutine, Counters & time delay, Code conversion, BCD arithmetic, 16 bit data operation.

UNIT III

Interrupt & Interfacing some peripheral I/O: Interfacing data converters, Programmable Interface Devices: 8155 I/O and Timer, 8279 Keyboard / Display interface

UNIT IV

General purpose programmable peripheral devices: 8255 (Bidirectional data transfer between two computer) 8254 (Programmable Interval Timer), 8259A Interrupt Controller, 8237 DMA, Serial I/O Communication.

UNIT V

Other eight bit, sixteen-bit Microprocessor: Z80, MC-6800, MC-68000, NSC

Introduction to advance Microprocessor: 8086, 80286, 80386, Microcontroller 8051.

Text Books:

1. R.S. Gaonkar, Microprocessor Architecture Programming and Application of 8085(Latest Edition).

Reference Book:

1. Shridhar and Ghosh, 0000 to 8085 Microprocessor.
2. Intel Corporation, Microprocessors and peripheral hand book.

INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, DAVV, INDORE
M. Tech. (IT) 5 Years V SEMESTER
IT-505B: Programming in Java

Aim of Course: To learn the Java programming language fundamentals: its syntax, idioms, patterns, and styles with object oriented programming concepts.

Objectives:

The course is designed to make students:

- Write programs using the Java language. Basic topics considered are programs and program structure in general, and Java syntax, data types, flow of control, classes, methods, objects, arrays, exception handling, recursion, and graphical user interfaces (GUIs).

Course Contents:

UNIT I

Introduction to Java: Features of Java, Object-oriented programming overview, Introduction of Java Technologies, How to write simple Java programs, Data Types, Variables, Memory concepts, decision making operators, Naming Conventions Introduction to Class, Objects, Methods and Instance Variables, Primitive type Vs Reference Type, Initializing Objects with Constructors. Type conversion & casting, Operators, Control statement, while, do-while, for, foreach Statements, switch Multiple-Selection Statement, break and continue Statements. Static Method, static field and Math Class, Argument Promotion and Casting, Scope of declaration and Method Overloading.

String Handling & Arrays: String Handling: The String constructors, String operators, Character Exaction, String comparison, String Buffer.

Arrays: Declaring and Creating Arrays, Passing Arrays to Method, Multidimensional Arrays, Variable-Length Argument lists, Using Command-line Arguments. Final Instance Variables, this reference, static import, overloaded Constructors, Garbage collection and method finalize, Overloading methods, Parameter passing.

UNIT II

Inheritance & Polymorphism: Inheritance: Extending classes, protected Members, relationship between Superclasses and Subclasses, Using super, Constructor in Subclasses

Polymorphism: Method overriding, upcasting, Dynamic Method Dispatch, final Method and classes, Abstract classes and Methods, instanceof operator, Downcasting

Packages and Interfaces: Packages: Defining a Package, Understanding CLASSPATH, Access Protection, Importing packages, creating own packages.

Interfaces: Defining an Interface, Properties of interface, advantages of interface, Achieving multiple inheritance through interfaces, Variables in Interfaces.

UNIT III

Nested Classes & Exception Handling: Nested Classes: Overview of nested class and interfaces, static nested class and interfaces, non-static nested class and, anonymous classes.

Exception Handling: Introduction, overview of doing it and keywords used, when to use it, Java Exception Hierarchy, finally block, chained exceptions, declaring new exception types .

Streams and Files: Introduction to Data Hierarchy, Files and Streams, Sequential-access Text Files, Object Serialization, Random-Access files, Java Stream class Hierarchy.

UNIT IV

Multithreading: What are threads, The java thread model, Thread priorities, Thread life cycle, Creating thread and executing thread, Thread Synchronization, producer-consumer problem without Synchronization. Producer-consumer problem with Synchronization, Other class and Interfaces in java.util.concurrent, Monitor and Monitor Locks, Thread Groups, Synchronization, Inter-thread Communication.

Introduction to GUI & Applets: Introduction To GUI : Introduction, Overview of swing Components, Displaying text and Images in a window, Introduction to Event Handling, Common GUI Event Type and Listener Interfaces, How Event Handling Works, Adapter Classes, Layout Managers

Applets: Applet basics, Applet Architecture, Applet life cycle methods, Applet HTML Tag and attributes, Executing applet in web browser and in the appletviewer, in Passing parameters to Applets, doing GUI programming in applet.

UNIT V

Generic & Collection: API Generic: Introduction, Motivation for Generic Methods, Generic Methods : Implementation and Compile- time Translation Issues, Overloading Generic Methods, Generic Classes, Raw Types, Generic and Inheritance

Database connectivity: JDBC, The design of JDBC, Typical uses of JDBC, The Structured Query language, Basic JDBC Programming concepts, Executing Queries.

Text Books:

1. Deitel & Deitel, JAVA How to Program, Pearson Education, Sixth Edition
2. Herbert Schildt , Java : The Complete Reference, Tata McGraw- Hill, 7th Edition

Reference Books:

1. John Hubbard , Programming with Java (Schaum's Easy Outline)
2. JAVA 2 Black Book
3. Bruce Eckel , Thinking in Java, Prentice Hall
4. Gary Cornell, Cay Horstmann Core Java: Volume 1 Fundamentals, Eighth Edition, Pearson,
5. Sams Teach Yourself Java6 in 21 Days

INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, DAVV, INDORE
M. Tech. (IT) 5 Years V SEMESTER
IT-511: System Analysis & Design

Aim of Course: To introduce established and evolving methodologies for the analysis, design, and development of an information system.

Objectives: The course is designed to make students:

- Understand system characteristics, project management, prototyping, and systems development life cycle phases.
- To analyze a problem and design an appropriate solution using a combination of tools and techniques.

Prerequisite(s): Knowledge of following concepts is required, computer applications and software's, computer programming, Database management systems.

Course Contents:

UNIT- I

Overview of system analysis and design: Systems concepts, Definition, Characteristics of a system, Elements of a system, Types of System Physical or Abstract System, Open or Closed Systems, Man-Made Information Systems: Categories of Information, Formal Information Systems, Informal Information Systems.

UNIT- II

System Development Life Cycle: Recognition of need, feasibility study, Analysis, Design, Implementation, Post implementation and Maintenance, Project Termination, Prototyping.

Role of the system Analyst: Definition, Skills, Academic and Personal Qualifications, The Multifaceted Role of Analyst.

UNIT- III

System Analysis:

System Planning and the Initial Investigation: Bases of Planning in System Analysis, Dimensions of Planning, Initial Investigation, Needs Identification, Strategies for Determining Information Requirements, Problem Definition and Project Initiation, Background Analysis: Fact-Finding, Fact Analysis, Determination of Feasibility.

Structured Analysis: Introduction, Tools of Structured Analysis: Dataflow Diagrams, Data Dictionaries, Decision Tables, Decision Trees, Structured English.

Feasibility study: Introduction, Feasibility Considerations, Feasibility Study Stages, Feasibility Report, Cost/Benefit Analysis.

UNIT- IV

System design:

The Process and Stages of System Design: Introduction, The Process of Design: Logical and Physical Design, Design Methodologies: Structured Design, Form-Driven Methodology – The IPO Charts.

Input /Output and Form Design: Introduction, Input Design, Output Design, Form Design.

File Organization and Data Base Design: Introduction, File Structure, File Organization, Data Base Design, Views of Data, Data Structure.

UNIT- V

System Implementation, Post Implementation and Maintenance: Introduction, Testing objectives, System Testing, Types of System Tests, Quality Assurance: Quality Factors Specifications, Levels of Quality Assurance, Post Implementation and Maintenance, Project Scheduling, Project management.

Text Books:

1. System Analysis and Design by Elias M. Awad (GALGOTIA Publications)

Reference Books:

1. Analysis and Design of Information Systems by V. Rajaraman (PHI Publications)
2. System Analysis and Design & MIS by Anurag Jain (EXCEL BOOKS Publications)

INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, DAVV, INDORE

M.Tech. (IT) 5 Years V SEMESTER

Subject Code: IT-512 Subject Name: Discrete Structure

Aim of Course: To familiarize the students with mathematical concepts that underline much of computer science, and to help them develop the skills to solve problems using them, whether they are in a more advance course, doing research, or working.

Objectives:

The course is designed to make students:

1. Enhance mathematical reasoning of students.
2. To understand Discrete Mathematics such as sets, permutations, relations, graphs, trees and finite-state machines.
3. Enhance algorithmic thinking of students.

Course Contents:

UNIT I

Set theory: Introduction, sets and elements, universal set and empty set, subsets, Multi-set, Countable and uncountable sets, Venn diagrams, Set operations, Algebra of sets, Power sets, Partitions, Inclusion and exclusion, Mathematical induction, Ordered pair, Cartesian product, Computer representation of sets.

UNIT II

Relations: Introduction to relations, Pictorial representation of relations, Domain and range, Types of relations, Composition of relations, Equivalence relations, partially ordered relations. *Functions:* Introduction to functions, functions in terms of ordered pairs, Pictorial representation of functions, Types of functions: surjective, bijective, injective, etc., Inverse function, Equality of functions, Composition of functions.

UNIT III

Logic: Propositions and logic operations, Existential and universal quantifiers, Tautologies, Contradiction, Contingency, Logical equivalence.

Boolean algebra: Combinatorial circuits and their properties, Boolean functions and synthesis of circuits. *Lattices:* Partially ordered sets, Chains and anti chains, Representation and construction of Hasse diagrams, Special elements in POSETs, Lattices.

UNIT IV

Graph Theory-I: Definition and applications, Finite and infinite graphs, Incidence and degree, Isolated vertex, Pendent vertex, Types of graph, Subgraphs and isomorphic graph, Operations of graph, Paths, Cycles and connectivity, Eulerian and Hamiltonian graph, Planar graphs, Trees, Fundamental properties of trees, rooted and binary trees, spanning trees, fundamental circuits.

UNIT V

Graph theory-II: Cut sets and their properties, connectivity and separability, Network flows, 1 and 2 isomorphism, Matrix representation of graphs: Incidence and adjacency matrices, Diagraphs and shortest path algorithms, Applications of graphs, General discussion.

Reference Books:

1. J.P.Tremblay and R. Manohar . Discrete mathematical structures with applications to computer science, Tata McGraw Hill Publication
2. C.L.Liu . Elements of Discrete Mathematics, Tata McGraw Hill Publication
3. Llipschutz and Lipson. Discrete Mathematics, Schaum's outline series, Tata McGraw Hill Publication
4. K.A.Ross . Discrete Mathematics.
5. Bernard Kolman & Robert C. Busby. Discrete mathematical structures for Computer Science

INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, DAVV, INDORE
M. Tech. (IT) 5 Years V SEMESTER
IT-507C Programming in Java Lab Assignment

| SNo. | Program related to UNIT -I |
|------|--|
| 1. | Write a program that produces the following output: Hello World! It's been nice knowing you. Goodbye world! |
| 2. | State the order of evaluation of the operations in each of the following Java statements and implement them to show the value of x after each statement. (1) $x = 7 + 3 * 6 / 2 - 1$; (2) $x = 2 \% 2 + 2 * 2 - 2 / 2$; (3) $x = (3 * 9 * (3 + (9 * 3 / (3))))$; |
| 3. | Write an application that declares 5 integers, determines and prints the largest and smallest in the group. |
| 4. | Write an application that declares 5 integers, calculates and print the average of these numbers. |
| 5. | Write an application that declares two integers, determines whether the first is a multiple of the second and print the result. [Hint : Use the remainder operator.] |
| 6. | Write an application that calculates the product of the odd integers from 1 to 15. |
| 7. | Write an application that evaluates the factorial of the integers from 1 to 5. |
| 8. | Write an application Program to demonstrate your first object in java. |
| 9. | Write an application Program to demonstrate all control statements(selection,iteration and transfer). |

| | |
|-----|---|
| 10. | Write an application Program to demonstrate Method call activation records. |
| 11. | Write an application Program to demonstrate Method overloading. |
| 12. | Write an application for calculating Compound-interest (interest rate of 5% for 10 years) with for loop. |
| 13. | Write an application for demonstrating with for-each loop. |
| 14. | Write an application for demonstrate all shift operators. |
| 15. | Modify the above compound-interest application to repeat its steps for interest rates of 5, 6, 7, 8, 9 and 10%. Use a for loop to vary the interest rate. |
| 16. | Write an application Program of factorial number. |
| 17. | Write an application Program of fibonacci series. |
| 18. | Write an application Program of armstrong number. |
| 19. | WAP to determine whether an entered number is prime or not. |
| 20. | WAP in java to implement Selection Sort Algo. |
| 21. | WAP in java to implement Bubble Sort Algo. |
| 22. | WAP in java to implement Binary Search Algo. |
| 23. | WAP in java to demonstrate Stack class. |
| 24. | WAP in java to demonstrate Anonymous Array. |
| 25. | Write a Java program that randomly fill a 3 by 4 by 6 array and then prints the largest and smallest values in the array. |
| 26. | WAP in java to demonstrate 3D Array. |
| 27. | WAP in java to demonstrate VARANGS. |
| 28. | WAP in java to demonstrate Scanner class. |
| 29. | WAP in java to demonstrate Enumarated data type. |
| 30. | Write an application that uses String method equals and equalsIgnoreCase to tests any two string objects for equality. |
| 31. | Write an application that uses String method indexOf to determine the total number of occurrences of any given alphabet in a defined text. |
| 32. | Write an application that uses String method concat to concatenate two defined strings. |
| 33. | Write an application that finds the length of a given string. |
| 34. | Write an application that uses String method charAt to reverse the string. |
| 35. | Write an application that finds the substring from any given string using substring method and startsWith & endsWith methods. |
| 36. | Write an application that changes any given string with uppercase letters, displays it , changes |

| | |
|-----|--|
| | it back to lowercase letters and displays it. |
| | Program related to UNIT -II |
| 37. | Create a class called Employee that includes three pieces of information as instance variables – a first name (type String), a last name (type String) and a monthly salary (double) |
| 38. | Create a constructor in above class to initialize the three instance variables. Provide a get method for each instance variable. |
| 39. | Write a test application named EmployeeTest that demonstrates class Employee's capabilities. Create two employee objects and display each object's yearly salary. |
| 40. | Give each employee a 10% raise and display each Employee's yearly salary again. |
| 41. | Create a class Account with an instance variable balance (double). It should contain a constructor that initializes the balance, ensure that the initial balance is greater than 0.0. |
| 42. | Create two methods namely credit and getBalance. The first one adds the amount (passed as parameter) to balance and does not return any data. The second method allows clients (i.e. the other classes that use this class) to obtain the value of a particular Account object's balance. |
| 43. | Create class AccountTest to create and manipulate an Account object. |
| 44. | Write another method debit in the above program that withdraws money from an Account. Ensure that the debit amount does not exceed the Account's balance. In that case the balance should be left unchanged and the method should print a message indicating "Debit amount exceeded account balance". Modify class AccountTest to test method debit. |
| 45. | Write an application that reads a five digit integer and determine whether it is a palindrome (digit that reads the same backward and forward eg. 12321, 45554 etc.) . display an error message, if the number is no5 five digits long and allow the user to enter a new value. |
| 46. | Write an application that reads three nonzero value entered by the user and determines and prints sum, product, average, smallest & largest of three. |
| 47. | Write an application that prompts the user for the radius of a circle and uses a method called circleArea to calculate the area of the circle. |
| 48. | Add another method in the above program circlePerimeter to calculate the perimeter of the circle. |
| 49. | Write an application to create a super class Employee with information first name & last name and methods getFirstName(), getLastName() derive the sub-classes ContractEmployee and RegularEmployee with the information about department, designation & method displayFullName() , getDepartment, getDesig() to print the salary and to set department name |

| | |
|-----|---|
| | & designation of the corresponding sub-class objects respectively. |
| 50. | Derive sub-classes of ContractEmployee namely HourlyEmployee & WeeklyEmployee with information number of hours & wages per hour, number of weeks & wages per week respectively & method calculateWages() to calculate their monthly salary. Also override getDesig () method depending on the type of contract employee. |
| 51. | Write an application to create a super class Vehicle with information vehicle number,insurance number,color and methods getConsumption() and displayConsumption(). Derive the sub-classes TwoWheeler and FourWheeler with method maintenance() and average() to print the maintenance And average of vehicle. |
| 52. | Extend the above TwoWheeler class with methods getType() and getName() which gives the information about the type and the name of the company.Create sub-classes Geared and NonGeared with method average() to print the average of a geared and non-geared two wheeler. |
| 53. | Create a super class CommunityMember with the information of member i.e. name, address, contact, date_of_join, through methods getName (), getAddress (), getContact (), getDate_of_Join () and derive sub-classes Employee and Student with method Qualification () to print the related information with his/her qualification. |
| 54. | Create a super class Shape with methods getName() which gives the information about the type of the shape.derive its sub-classes TwoDim and ThreeDim with method area() and volume() respectively which prints the area and volume of a two-dimensional and three-dimensional shape. |
| 55. | Extend the class TwoDim with methods getLength(),getBreadth() which displays the length and breadth of two dimensional shapes.Derive sub-classes rectangle, rhombus with method getArea() and getPerimeter() to calculate the area and perimeter of this two dimensional shapes. . |
| 56. | Extend the class ThreeDim with methods getLength(),getBreadth(),getHeight() which displays the length , breadth and height of three dimensional shapes.Derive sub-classes cuboid,tetrahedron with method getArea() and getVolume() to calculate the area and volume of this three dimensional shapes. . |
| 57. | Create a super class Student with methods getQual (), getFirstName(),getLastName(), getAddress(), getContat(), which gives basic details of student.derive sub-classes Faculty and Scholar with method salary(), Course() resp. which gives the additional information about the salary and course of faculty and scholar resp. . |
| 58. | Create an abstract class Shape which calculate the area and volume of 2-d and 3-d shapes with methods getArea and getVolume. Reuse this class to calculate the area and volume of |

| | |
|------------------------------|---|
| | square ,circle ,cube. |
| 59. | Create an abstract class Employee with methods getAmount() which displays the amount paid to employee. Reuse this class to calculate the amount to be paid to WeeklyEmployee and HourlyEmployee according to no. of hours and total hours for HourlyEmployee and no. of weeks and total weeks for WeeklyEmployee. |
| 60. | Create an Interface payable with method getAmount ().Calculate the amount to be paid to Invoice and Employee by implementing Interface. |
| 61. | Create an Interface Vehicle with method getColor(),getNumber(),getConsumption(). Calculate the fuel consumed, name and color for TwoWheeler and FourWheeler by implementing interface Vehicle. |
| 62. | Create an Interface Fare with method getAmount() to get the amount paid for fare of travelling. Calculate the fare paid by bus and train implementing interface Fare. |
| 63. | Create an Interface StudentFee with method getAmount(), getFirstName(),getLastName(), getAddress(), getContact(). Calculate the amount paid by the Hostler and NonHostler student by implementing interface StudentFee |
| 64. | WAP to create your own package. Package should have more than two classes. Write a class that uses the package. |
| 65. | Create a package named org.shapes. Create some classes in the package representing some common geometric shapes like Square, Triangle, Circle and so on. |
| Program related to UNIT -III | |
| 66. | Exception Handling program for division of two numbers that accepts numbers from user. |
| 67. | Exception Handling program for storing values in array of int or String that results into buffer overflow |
| 68. | Exception Handling program for NullPointerException--thrown if the JVM attempts to perform an operation on an Object that points to no data, or null |
| 69. | Exception Handling program for NumberFormatException--thrown if a program is attempting to convert a string to a numerical datatype, and the string contains inappropriate characters (i.e. 'z' or 'Q') |
| 70. | Exception Handling program for ClassNotFoundException--thrown if a program can not find a class it depends at runtime (i.e., the class's ".class" file cannot be found or was removed from the CLASSPATH) |
| 71. | Exception Handling program for IOException--actually contained in java.io, but it is thrown if the JVM failed to open an I/O stream |

| | |
|-----------------------------|---|
| 72. | Write a program that shows that the order of the catch blocks is important. If you try to catch a superclass exception type before a subclass type, the compiler should generate errors. |
| 73. | Program for demonstrating the use of throw, throws & finally - Create a class with a main() that throws an object of class Exception inside a try block. Give the constructor for Exception a String argument. Catch the exception inside a catch clause and print the String argument. Add a finally clause and print a message to prove you were there. |
| 74. | Create your own exception class using the extends keyword. Write a constructor for this class that takes a String argument and stores it inside the object with a String reference. Write a method that prints out the stored String. Create a try-catch clause to exercise your new exception. |
| 75. | Write a program to rethrow an exception – Define methods one() & two(). Method two() should initially throw an exception. Method one() should call two(), catch the exception and rethrow it Call one() from main() and catch the rethrown exception. |
| 76. | Write a program to change the priority of thread. |
| 77. | WAP for producer consumer problem (with synchronization). |
| 78. | Open a text file so that you can read the file one line at a time. Read each line as a String and send the results to System.out. |
| 79. | Modify Exercise 1 so that the name of the file you read is provided as a command-line argument. |
| 80. | Modify Exercise 2 to force all the lines in the results to upper case and send the results to System.out |
| 81. | Modify Exercise 2 to also open a text file so you can write text into it. |
| Program related to UNIT -IV | |
| 82. | Create an application to draw a horizontal line. |
| 83. | Create an application to draw one line perpendicular to other. One line parallel to other. |
| 84. | Create an application to display a circle within rectangle |
| 85. | In the above application fill different colors in the circle & rectangle. |
| 86. | Write an application that displays any string. Choose color from combo box to change the color of this displayed string. |
| 87. | WAP to demonstrat AWT buttons with event handling. |
| 88. | WAP to demonstrat BorderLayout in AWT window. |
| 89. | WAP in java to create a file. |
| 90. | WAP in java to delete a file. |

| | |
|------|--|
| 91. | WAP in java to determine a file or dir exist or not. |
| 92. | WAP in java to determine whether a file or dir. |
| 93. | WAP in java to exit from a Frame window when we click on Close button. |
| 94. | WAP in java to move a file. |
| 95. | WAP in java to read a file using FileReader and break the contents using StringTokenizer. |
| 96. | WAP in java to write into a file using FileWriter. |
| 97. | WAP in java to demonstrate RandomAccessFile. |
| 98. | WAP in java to rename a file. |
| 99. | WAP in java to get file length. |
| 100. | WAP in java to reverse a string by word. |
| 101. | WAP in java to demonstrate StringTokenizer class. |
| 102. | WAP to create your own package in which you define a class StringUtils in this package. |
| 103. | WAP in java to create a thread that prints counting by extending Thread class. |
| 104. | WAP in java to demonstrate current thread. |
| 105. | WAP in java to create a thread using Runnable interface. |
| 106. | WAP in java to determine whether a thread is alive or not. |
| 107. | WAP in java to demonstrate join() method of Thread class. |
| 108. | WAP in java to implement toString() method in your class to print objects. |
| | Program related to UNIT -V |
| 109. | WAP to change background color according to selected color from combo box. |
| 110. | Write a program in java to scroll a string using Applet. |
| 111. | WAP in java to demonstrate all mouse events. |
| 112. | WAP in java to demonstrate all keyboard events. |
| 113. | WAP in java to demonstrate GridLayout. |
| 114. | WAP in java to create an Applet having status bar. |
| 115. | WAP in java to demonstrate Color class. |
| 116. | WAP in java to event handling using adapter class. |
| 117. | WAP in java to demonstrate Check Boxes in applet. |
| 118. | WAP in java to demonstrate Calendar class. |
| 119. | WAP in java to display all available fonts in an applet window |
| 120. | WAP to copy the content of a file to another file. |
| 121. | WAP in java to demonstrate Random class. |
| 122. | WAP in java to create an Applet that's background color will be change on each second. |
| 123. | WAP in java to sum two 2-D matrix and store the result into third matrix. |
| 124. | Create a program that will print every argument given on the command line. consider how your program will deal with no argument. |
| 125. | WAP to draw a string and choose its size respectively from combo box. |
| 126. | WAP in java to demonstrate data entry program. |
| 127. | Create an application of cash withdrawal from the bank account that have no. of users that are operating the accounts.(synchronization) |
| 128. | WAP to create three text boxes and save entered value into a file. |
| 129. | Implement a class Reader that count the number of times a particular character, such as e, is |

| | | |
|------|--|---|
| | | |
| | | read. The character can be specified when the stream is created. |
| 130. | | Construct a program Wc ("word count"), which counts number of chars, words and lines of the text file. Space is counted as a character. Empty rows are counted as lines. "Word" will represent a string. |
| 131. | | Write a small application with a default date 01/01/2000 and three combo boxes displaying valid days, months & year(1990 – 2050). Change the displayed date with the one chosen by user from these combo boxes. |
| 132. | | Create a GUI with a text field and three buttons. When you press each button, make some different text appear in the text field. |
| 133. | | Create a GUI application to take input of two numbers(text field) from user. When you press button it should display sum of the two numbers in a third text box. |
| 134. | | Create an applet with a Button and a TextField. Write a referenceEvent() so that if the button has the focus, characters typed into it will appear in the TextField. |
| 135. | | Write an application to create a GUI with two buttons such that clicking on the first displays the message “Welcome to SCS” on the window and clicking on the second changes the color of the message(hint : toggle the color) |
| 136. | | Create a GUI with title STUDENT which has labels roll no., name, class, address with textboxes for taking input from the user(without any functionality). |
| 137. | | Create a GUI application for fees receipt which contains checkboxes for selecting the course, radio buttons for selecting gender and labels and corresponding textboxes for name, class, date and amount paid. |
| 138. | | Create a GUI application to display a calculator using grid Layout (You do not have to provide functionality). |
| 139. | | WAP that generate a random number (1 – 10000). Let the user guess the correct number. User will enter the digit. Program should let the user that input is right or wrong. No of turns that user can make a choice for input is twice the number of digits in the system generated numbers. |
| 140. | | Convert the input date in words. Input format is dd mm yy. |

INTERNATIONAL INSTITUTE OF PROFESSIONAL STUDIES, DAVV, INDORE
M.Tech. (IT) 5 Years V SEMESTER

IT-508E Microprocessor and Assembly Language Programming Lab Assignment

1. Exchange of two variables through

(a) Memory Location (b) Two Register (c) Register and Memory location

2. Program to add 2 numbers from

(a) Two Memory Location (b) Two Register (c) Register and Memory location

3. Program to subtract 2 numbers from

(a) Two Memory Location (b) Two Register (c) Register and Memory location

4. Program to OR contents of register and accumulator.

5. Program to XOR contents of register and accumulator.

6. Program to check equality of 2 numbers.

7. Program to divide decimal 42 by 5 and store result in register D.

8. Program to multiply decimal 04 and 05.

9. Program to generate Fibonacci series at memory location from 2050 to 2059.

10. Program to find square root of decimal 36.

11. Program to find factorial of any number.

12. Program to find minimum of 10 numbers.

13. Program to find maximum of 10 numbers.

14. Program to add contents of memory location 2050 and 2051 and store result at 2090.

15. Program to find minimum of 2 numbers.

16. Program to AND contents of register and accumulator.

17. Program to multiply 2 decimal numbers in which result is greater than 8 bits.

18. Program to check 4th bit of a 8 bit number.

19. Program to add two 16 bit numbers.

20. Program to find summation ($n*n$) where n varies from 1 to 8.

21. Program to add 2 BCD numbers.

22. Program to arrange numbers in ascending order stored which are at memory location: 2050 to 2059.

23. Program to arrange numbers in descending order which are stored at memory location: 2050

to 2059.

24. Program to subtract two 16 bit numbers.
25. Program to demonstrate use of ADC.
26. Program to exchange contents of DE and HL register pair.
27. Program to implement UP counter.
28. Program to implement DOWN counter.
29. Program to convert a number from hexadecimal to binary.
30. Program to implement above question with help of sub routine.
31. Program to compliment contents of the accumulator.
32. Program to find smallest element in the array.
33. Program to find largest element in the array.
34. Program to arrange array in ascending order.
35. Program to arrange array in descending order.