JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR
Common Course Structure for

Computer Science & Engineering B. Tech Course (2015-16)

II B. Tech – II Sem

S.No.	Course Code	Subject	Tu	T	Lab	C
1	15A54401	Probability and Statistics	3	1	-	3
2	15A05401	Software Engineering	3	1	-	3
3	15A05402	Computer Organization	3	1	-	3
4	15A04407	Microprocessors & Interfacing	3	1	-	3
5	15A05403	Object Oriented Programming using Java	3	1	-	3
6	15A05404	Formal Languages and Automata Theory	3	1	-	3
7	15A04408	Microprocessors & Interfacing Laboratory	-	-	4	2
8	15A05405	Java Programming Laboratory	-	-	4	2
9	15A05406	Comprehensive Online Examination-I	-	-	-	1
		Total	**18**	**06**	**08**	**23**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)	T	Tu	C
	3	1	3

(15A54401) PROBABILITY AND STATISTICS

(Common to CSE, IT, Civil, Mech.)

Objectives: To help the students in getting a thorough understanding of the fundamentals of probability and usage of statistical techniques like testing of hypothesis, Statistical Quality Control and Queuing theory

UNIT – I

Basic concepts of Probability – Random variables – Expectation – Discrete and continuous Distributions – Distribution functions. Binomial and poison distributions Normal distribution – Related properties.

UNIT – II

Test of Hypothesis: Population and Sample - Confidence interval of mean from Normal distribution - Statistical hypothesis - Null and Alternative hypothesis - Level of significance. Test of significance - Test based on normal distribution - Z test for means and proportions.

UNIT – III

Small samples - t- test for one sample and two sample problem and paired t-test, F-test and Chi-square test (testing of goodness of fit and independence).

UNIT – IV

Statistical Quality Control: Concept of quality of a manufactured product -Defects and Defectives - Causes of variations - Random and assignable - The principle of Shewhart Control Chart-Charts for attribute and variable quality characteristics- Constructions and operation of $\overline{X}$ - Chart, R-Chart,

p - Chart and C-Chart.

UNIT – V

Queuing Theory: Pure Birth and Death process, M/M/1 & M/M/S & their related simple problems.

TEXT BOOKS:

1. Probability & Statistics by E. Rukmangadachari & E. Keshava Reddy, Pearson Publisher.
2. Probability & Statistics for engineers by Dr. J. Ravichandran WILEY-INDIA publishers.

REFERENCES:

1. Probability & Statistics by T.K.V. Iyengar, B.Krishna Gandhi, S.Ranganatham and M.V.S.S.N.Prasad, S.Chand publications.
2. Statistical methods by S.P. Gupta, S.Chand publications.
3. Probability & Statistics for Science and Engineering by G.Shanker Rao, Universities Press.
4. Probability and Statistics for Engineering and Sciences by Jay L.Devore, CENGAGE.

5. Probability and Statistics by R.A. Jhonson and Gupta C.B.

Outcomes: The student will be able to analyze the problems of engineering & industry using the techniques of testing of hypothesis, Statistical Quality Control and Queuing theory and draw appropriate inferences.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)	T	Tu	C
	3	1	3

(15A05401) SOFTWARE ENGINEERING

Course Objectives
*   To understand the software life cycle models.
*   To understand the software requirements and SRS document.
*   To understand the importance of modeling and modeling languages.
*   To design and develop correct and robust software products.
*   To understand the quality control and how to ensure good quality software.
*   To understand the planning and estimation of software projects.
*   To understand the implementation issues, validation and verification procedures.
*   To understand the maintenance of software

Course Outcomes
*   Define and develop a software project from requirement gathering to implementation.
*   Ability to code and test the software
*   Ability to plan, Estimate and Maintain software systems

Unit I:
Software and Software Engineering: The Nature of Software, The Unique Nature of WebApps, Software Engineering, The Software Process, Software Engineering Practice, Software Myths
Process Models: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Technology, Product and Process.
Agile Development: Agility, Agility and the Cost of Change, Agile Process, Extreme Programming, Other Agile Process Models
Unit II:
Understanding Requirements: Requirements Engineering, Establishing the groundwork, Eliciting Requirements, Developing Use Cases, Building the requirements model, Negotiating Requirements, Validating Requirements.
Requirements Modeling (Scenarios, Information and Analysis Classes): Requirements Analysis, Scenario-Based Modeling, UML Models that Supplement the Use Case, Data Modeling Concepts, Class-Based Modeling.
Requirements Modeling (Flow, Behavior, Patterns and WEBAPPS): Requirements Modeling Strategies, Flow-Oriented Modeling, Creating a Behavioral Model, Patterns for Requirements Modeling, Requirements Modeling for WebApps.

Unit III:
Design Concepts: Design with Context of Software Engineering, The Design Process, Design Concepts, The Design Model.
Architectural Design: Software Architecture, Architecture Genres, Architecture Styles, Architectural Design, Assessing Alternative Architectural Designs, Architectural Mapping Using Data Flow.
Component-Level Design: Component, Designing Class-Based Components, Conducting Component-level Design, Component Level Design for WebApps, Designing Traditional Components, Component-Based Development.

Unit IV:
User Interface Design: The Golden Rules, User Interface Analysis and Design, Interface Analysis, Interface Design Steps, WebApp Interface Design, Design Evaluation.
WebApp Design: WebApp Design Quality, Design Goal, A Desigin Pyramid for WebApps, WebApp Interface Design, Aestheic Design, Content Design, Architecture Design, Navigation Design, Component-Level Design, Object-Oriented Hypermedia Design Method(OOHMD).

Unit V:
Software Testing Strategies: A strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, Test Strategies for Object-Oriented Software, Test Strategies for WebApps, Validation Testing, System Testing, The Art of Debugging.
Testing Conventional Applications: Software Testing Fundamentals, Internal and External Views of Testing, White-Box Testing, basic Path testing, Control Structure Testing, Black-Box Testing, Model-based Testing, Testing for Specialized Environments, Architectures and Applications, Patterns for Software Testing.Testing Object-Oriented Applications: Broadening the View of Testing, Testing with OOA and OOD Models, Object-Oriented Testing Strategies, Object-Oriented Testing Methods, Testing Methods Applicable at the Class level, Interclass Test-Case Design.

Textbook:
1. "Software engineering A practitioner's Approach", Roger S. Pressman, McGraw Hill International Education, Seventh Edition, 2016.

Reference Textbooks:
1. Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI,
2. Software Engineering, Ninth Edition, IAN Sommerville, Pearson, Ninth edition.
3. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India,2010.
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition, 2006.
7. Software Engineering Foundations, Yingxu Wang, Auerbach Publications,2008.
8. Software Engineering Principles and Practice, Hans Van Vliet,3rd edition, John Wiley &Sons Ltd.

9. Software Engineering 3: Domains, Requirements, and Software Design, D.Bjorner, Springer International Edition.

10. Introduction to Software Engineering R.J.Leach, CRC Press

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)

	T	Tu	C
	3	1	3

(15A05402) COMPUTER ORGANIZATION

Course Objectives:

- To learn the fundamentals of computer organization and its relevance to classical and modern problems of computer design
- To make the students understand the structure and behavior of various functional modules of a computer.
- To understand the techniques that computers use to communicate with I/O devices
- To study the concepts of pipelining and the way it can speed up processing.
- To understand the basic characteristics of multiprocessors

Course Outcomes:

- Ability to use memory and I/O devices effectively
- Able to explore the hardware requirements for cache memory and virtual memory
- Ability to design algorithms to exploit pipelining and multiprocessors

Unit I:

Basic Structure of Computer: Computer Types, Functional Units, Basic operational Concepts, Bus Structure, Software, Performance, Multiprocessors and Multicomputer.

Machine Instructions and Programs: Numbers, Arithmetic Operations and Programs, Instructions and Instruction Sequencing, Addressing Modes, Basic Input/output Operations, Stacks and Queues, Subroutines, Additional Instructions.

Unit II:

Arithmetic: Addition and Subtraction of Signed Numbers, Design and Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations.

Basic Processing Unit: Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control, Multiprogrammed Control.

Unit III:

The Memory System: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage.

Unit IV:

Input/output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces.

Unit V:

Pipelining: Basic Concepts, Data Hazards, Instruction Hazards, Influence on Instruction Sets.

Large Computer Systems: Forms of Parallel Processing, Array Processors, The Structure of General-Purpose, Interconnection Networks.

Textbook:

1) "Computer Organization", Carl Hamacher, Zvonko Vranesic, Safwat Zaky, McGraw Hill Education, 5th Edition, 2013**.**

Reference Textbooks:

1. Computer System Architecture, M.Morris Mano, Pearson Education, 3rd Edition.
2. Computer Organization and Architecture, Themes and Variations, Alan Clements, CENGAGE Learning.
3. Computer Organization and Architecture, Smruti Ranjan Sarangi, McGraw Hill Education.
4. Computer Architecture and Organization, John P.Hayes, McGraw Hill Education.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)

	T	Tu	C
	3	1	3

(15A04407) MICROPROCESSORS & INTERFACING

Course Objective:
- *Study the instruction set of 8086 microprocessor and its architecture*
- *Learn assembly language programming using 8086 microprocessor*
- *Interfacing 8051, 8255, 8237, and 8259*

Learning Outcome:
- *Program the 8086 microprocessor*
- *Interface the 8086 microprocessor with various devices and program them*

UNIT I

Microprocessors-Evolution and Introduction: Microprocessors and Micro Controllers, Microprocessor based system, Origin of Microprocessor, Classification of Microprocessors, Types of Memory, I/O Devices, Technology Improvements Adapted to Microprocessors and Computers, Introduction to 8085 processor, Architecture of 8085, Microprocessor instructions, classification of instructions, Instruction set of 8085.

Intel 8086 Microprocessor architecture, Features, and Signals: Architecture of 8086, Accessing memory locations, PIN details of 8086.

UNIT II

Addressing Modes, Instruction Set and Programming of 8086: Addressing modes in 8086, Instruction set of 8086, 8086 Assembly Language Programming, Modular Programming.

UNIT III

8086 Interrupts: Interrupt types in 8086, Processing of Interrupts by 8086, Dedicated interrupt types in 8086, Software interrupts-types 00H-FFH, Priority among 8086 interrupts, Interrupt service routines, BIOS interrupts or functional calls, Interrupt handlers, DOS services-INT 21H, System calls-BIOS services.

Memory and I/O Interfacing: Physical memory organization in 8086, Formation of system bus, Interfacing RAM and EPROM chips using only logic gates, Interfacing RAM/ EPROM chips using decoder IC and logic gates, I/O interfacing, Interfacing 8-bit input device with 8086, Interfacing output device using 8086, Interfacing printer with 8086, Interfacing 8-bit and 16-bit I/O devices or ports with 8086, Interfacing CRT terminal with 8086.

UNIT IV

Features and Interfacing of programmable devices for 8086 systems: Intel 8255 programmable peripheral interface, Interfacing switches and LEDS, Interfacing seven segment displays, Traffic light control, Interfacing analog to digital converters, Intel Timer IC 8253, Introduction to serial communication, 8259 programmable controller, 8237 DMA controller.

UNIT V

Introduction to 8051 Micro controllers: Intel's MCS-51 series micro controllers, Intel 8051 architecture, Memory organization, Internal RAM structure, Power control in 8051, Stack operation.

8051 Instruction Set and Programming: Introduction, Addressing modes of 8051, Instruction set of 8051, Hardware features of 8051: Introduction, Parallel ports in 8051, External memory interfacing in 8051, Timers, Interrupts, Serial ports.

Interfacing Examples: Interfacing 8255 with 8051, Interfacing of push button switches and LEDS, Interfacing of seven segment displays.

Text Books:
- *"Microprocessor and Interfacing 8086,8051, 8096 and advanced processors", Senthil Kumar,*
  Saravanan, Jeevanathan, Shah, 1st edition, 2nd impression, 2012, Oxford University Press.
- *"The X86 Microprocessors", Lyla B. Das. , 2010, Pearson.*

Reference Books:
1. *"Microprocessor and Interfacing: Programming and Hardware", Douglas V.Hall, McGrawHill*
2. *"8086 microprocessor: Programming and Interfacing the PC", Kenneth Ayala, Cengage Learning*
3. *"ARM system-on-chip architecture", Steve Furber, Addison-Wesley Professional*
4. *"The Intel Microprocessors", Barry B. Brey, Prentice Hall*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)	T	Tu	C
	3	1	3

(15A05403) OBJECT ORIENTED PROGRAMMING USING THOUGH JAVA

Course Objectives:
- Study the syntax, semantics and features of Java Programming Language
- Learn the method of creating Multi-threaded programs and handle exceptions
- Learn Java features to create GUI applications & perform event handling

Course Outcomes:

- Ability to solve problems using object oriented approach and implement them using Java
- Ability to write Efficient programs with multitasking ability and handle exceptions
- Create user friendly interface

UNIT I:

The History and Evolution of Java:

Java's Lineage, The Creation of java, how java changed the internet, Java's magic: The byte code, Servlets: java on the server side, java Buzzwords, Evolution of java.

An Overview of Java:

Object Oriented Programming, Two control statements, Using blocks of codes, Lexical issues,
The java class Libraries.

Data Types, Arrays and Variables:

Primitive Types, Integers, Floating-point Types, Characters, Booleans, literals, variables, Type conversion and casting, Automatic Type Promotion in Expressions, Arrays, strings, Pointers.

UNIT II:

Operators:
Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logic operators, The assignment operator, The ? Operator, Operator Precedence, Using Parentheses.

Control Statements:
Java's selection Statements, Iteration statements, Jump Statements.

Introducing Classes:

Class Fundamentals, Declaring Objects, Assuming Object reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The Finalize() method, A Stack class. Overloading Methods, Using Object as Parameter, Argument Passing, Returning Objects, Recursion, Introducing Access control, Understanding static, Introducing Nested and Inner classes, Exploring the String class, Using Command line Arguments, Varargs: variable-Length Arguments.

UNIT III:

Inheritance: Basics, Using super, creating a multi level hierarchy, when constructors are executed, method overriding, dynamic method dispatch, using abstract class, using final with inheritance, the object class.

Packages and Interfaces:
Packages, Access protection, Importing Packages, Interfaces, Default Interfaces, Default interface methods, Use static methods in an Interface, Final thoughts on Packages and interface**s.**

Exception Handling:

Exception handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch,
Multiple catch clauses, Nested try statements, throw, throws, finally, Java Built-in Exceptions, Creating your own exception subclasses, Chained Exceptions, Three Recently added Exceptions features, Using Exceptions.

UNIT IV:
Multithreaded Programming:

The java Thread Model, The main thread , Creating Thread, Creating Multiple Threads, Using isAlive() and join(), Thread Priorities, Synchronization, Interthread Communication, Suspending, resuming and stopping threads, Obtaining a thread state, Using Multithreading.

I/O, Applets, and Other Topics:

I/O basics, Reading Console input, Writing console Output, The PrintWriter class, Reading and writing files, Automatically closing a file, Applet fundamentals, enumerations type wrappers auto boxing annotations, Generics: The general form of a generics class, creating a generic method, generics interfaces.

UNIT V:
Introduction the AWT: Working with windows, graphics and Text:
AWT classes, window fundamentals, working with frame windows, creating a frame window in a an AWT Based applet, creating a window program, displaying information within a window, Graphics, working with color, setting the paint mode, working with fonts, managing text output using font metrics,.
Using AWT controls, Layout Mangers, and Menus:
AWT control fundamentals, Labels, using buttons, applying check boxes, check box group, choice controls, using lists, Managing scroll bars, using a Text field, Using a Text area, understanding layout managers, Menu bars and Menus, dialog boxes, file dialog, Overriding paint().

TEXT BOOKS:
1."Java The Complete Reference", Herbert Schildt, MC GRAW HILL Education, 9th
Edition,2016.

REFENCE BOOKS:

1. "Programming with Java" T.V.Suresh Kumar, B.Eswara Reddy, P.Raghavan Pearson Edition.
2. "Java Fundamentals - A Comprehensive Introduction", Herbert Schildt and Dale Skrien, Special Indian Edition, McGrawHill, 2013.
3. "Java – How to Program", Paul Deitel, Harvey Deitel, PHI.
4. "Core Java", NageswarRao, Wiley Publishers.
5. "Thinking in Java", Bruce Eckel, Pearson Education.
6. "A Programmers Guide to Java SCJP", Third Edition, Mughal, Rasmussen, Pearson.

"Head First Java", Kathy Sierra, Bert Bates, O'Reilly "SCJP – Sun Certified Programmer for Java Study guide" – Kathy Sierra, Bert Bates, McGrawHill

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)   T   Tu   C
                                                                      3   1   3

(15A05404) FORMAL LANGUAGES AND AUTOMATA THEORY

Course Objective:

- *Understand formal definitions of machine models.*
- *Classify machines by their power to recognize languages.*
- *Understanding of formal grammars, analysis*
- *Understanding of hierarchical organization of problems depending on their complexity*
- *Understanding of the logical limits to computational capacity*
- *Understanding of undecidable problems*

Learning Outcome:

At the end of the course, students will be able to

- *Construct finite state diagrams while solving problems of computer science*
- *Find solutions to the problems using Turing machines*
- *Design of new grammar and language*

UNIT I

Introduction: Basics of set theory, Relations on sets, Deductive proofs, Reduction to definitions, Other theorem forms, Proving equivalences about sets, The Contrapositive, Proof by contradiction, Counter examples, Inductive proofs, Alphabets, Strings, Languages, Problems, Grammar formalism, Chomsky Hierarchy

Finite Automata: An Informal picture of Finite Automata, Deterministic Finite Automata (DFA), Non Deterministic Finite Automata (NFA), Applying FA for Text search, Finite Automata with

Epsilon transitions (ϵ-NFA or NFA- ϵ), Finite Automata with output, Conversion of one machine to another, Minimization of Finite Automata, Myhill-Nerode Theorem.

UNIT II

Regular Languages: Regular Expressions (RE), Finite Automata and Regular Expressions,

Applications of Regular Expressions, Algebraic laws for Regular Expressions, The Arden's Theorem, Using Arden's theorem to construct RE from FA, Pumping Lemma for RLs, Applications of Pumping Lemma, Equivalence of Two FAs, Equivalence of Two REs, Construction of Regular Grammar from RE, Constructing FA from Regular Grammar, Closure properties of RLs, Decision problem's of RLS, Applications of REs and FAs

UNIT III

Context Free Grammars and Languages: Definition of Context Free Grammars (CFG), Derivations and Parse trees, Ambiguity in CFGs, Removing ambiguity, Left recursion and Left factoring, Simplification of CFGs, Normal Forms, Linear grammars, Closure properties for CFLs, Pumping Lemma for CFLs, Decision problems for CFLs, CFG and Regular Language..

UNIT IV

Push Down Automata (PDA): Informal introduction, The Formal Definition, Graphical notation, Instantaneous description, The Languages of a PDA, Equivalence of PDAs and CFGs, Deterministic Push Down Automata, Two Stack PDA.

UNIT V

Turing Machines and Undecidability: Basics of Turing Machine (TM), Transitional Representation of TMs, Instantaneous description, Non Deterministic TM, Conversion of Regular Expression to TM, Two stack PDA and TM, Variations of the TM, TM as an integer function, Universal TM, Linear Bounded Automata, TM Languages, Unrestricted grammar , Properties of Recursive and Recursively enumerable languages, Undecidability, Reducibility, Undeciadable problems about TMs, Post's Correspondence Problem(PCP), Modified PCP.

Text Books:
  1. *Introduction to Automata Theory, Formal Languages and Computation, Shyamalendu kandar, Pearson.*
  2. *Introduction to Automata Theory, Languages, and Computation, Third Edition, John E.Hopcroft, Rajeev Motwani, Jeffery D. Ullman, Pearson.*

Reference Books:
  1. *Introduction to Languages and the Theory of Computation, John C Martin, TMH, Third Edition.*
  2. *Theory of Computation, Vivek Kulkarni, OXFORD.*
  3. *Introduction to the Theory of Computation., Michel Sipser, 2nd Edition, Cengage Learning*
  4. *Theory of computer Science Automata, Languages and Computation, K.L.P. Mishra, N. Chandrasekaran, PHI, Third Edition.*
  5. *Fundamentals of the Theory of Computation, Principles and Practice, Raymond Greenlaw, H. James Hoover, Elsevier, Morgan Kaufmann.*
  6. *Finite Automata and Formal Language A Simple Approach, A.M. Padma Reddy, Pearson*

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)   **L   C**
                                                                   4   2

(15A04408) MICRO PROCESSORS & INTERFACING LAB

Course Objective:
- *To become skilled in 8086 Assembly Language programming.*
- *To understand programmable peripheral devices and their Interfacing.*
- *To understand and learn 8051 microcontroller.*
- *To learn 8051 assembly Language programming*

Learning Outcome:
- *Able to write8086 Assembly Language programs.*
- *Able to understand programmable peripheral devices and their Interfacing.*
- *Able to write 8051 assembly Language programs.*

Minimum **Ten** Experiments to be conducted (**Five** from each section)

I)   8086 Microprocessor Programs using MASM/8086 kit.
1. Introduction to MASM Programming.
2. Arithmetic operation – Multi byte Addition and Subtraction, Multiplication and Division – Signed and unsigned Arithmetic operation, ASCII – arithmetic operation.
3. Logic operations – Shift and rotate – Converting packed BCD to unpacked BCD, BCD to ASCII conversion.
4. By using string operation and Instruction prefix: Move Block, Reverse string, Sorting, Length of the string, String comparison.

Interfacing:
1. 8259 – Interrupt Controller and its interfacing programs
2. 8255 – PPI and its interfacing programs (A /D, D/A, stepper motor,)
3. 7-Segment Display.

II)   Microcontroller 8051 Trainer kit
1. Arithmetic operation – Multi byte Addition and Subtraction, Multiplication and Division – Signed and unsigned Arithmetic operation.
2. Logic operations – Shift and rotate.
3. Sorting- Ascending and descending order.

Interfacing using 8051 Trainer kit:
1. Key board Interfacing
2. Seven Segment display
3. Switch Interfacing
4. Relay Interfacing
5. UART

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

B. Tech II - II sem (C.S.E)   **L   C**

                                                                   4   2

(15A05405) JAVA PROGRAMMING LABORATORY

Course Objectives:

- Learn to use object orientation to solve problems and use java language to implement them.
- To experiment with the syntax and semantics of java language and gain experience with java programming

Course Outcomes:

- Ability to write portable programs which work in all environments
- Ability to create user friendly interfaces
- Ability to solve the problem using object oriented approach and design solutions which are robust

List of Experiments

1) Preparing and practice – Installation of Java software, study of any Integrated development environment, sample programs on operator precedence and associativity, class and package concept, scope concept, control structures, constructors and destructors. Learn to compile, debug and execute java programs.
2) Write Java program(s) on use of inheritance, preventing inheritance using final, abstract classes.
3) Write Java program(s) on dynamic binding, differentiating method overloading and overriding.
4) Write Java program(s) on ways of implementing interface.
5) Write a program for the following
   - Develop an applet that displays a simple message.
   - Develop an applet for waving a Flag using Applets and Threads.
6) Write Java program(s) which uses the exception handling features of the language, creates exceptions and handles them properly, uses the predefined exceptions, and create own exceptions
7) Write java program that inputs 5 numbers, each between 10 and 100 inclusive. As each number is read display it only if it's not a duplicate of any number already read. Display the complete set of unique values input after the user enters each new value.

18

8) Write Java program(s) on creating multiple threads, assigning priority to threads, synchronizing threads, suspend and resume threads

10) Write a java program to split a given text file into n parts. Name each part as the name of the original file followed by .part<n> where n is the sequence number of the part file.

11) Write a java program to create a super class called Figure that receives the dimensions of two dimensional objects. It also defines a method called area that computes the area of an object. The program derives two subclasses from Figure. The first is Rectangle and second is Triangle. Each of the sub classes override area() so that it returns the area of a rectangle and triangle respectively.

12) Write a Java program that creates three threads. First thread displays "Good Morning" every one second, the second thread displays "Hello" every two seconds and the third thread displays "Welcome" every three seconds

13) Design a simple calculator which performs all arithmetic operations. The interface should look like the calculator application of the operating system. Handle the exceptions if any.

14) Write a java program to handle mouse events

15) Write a java program to handle keyboard events

16) Write a java program that allows conduction of object type examination containing multiple choice questions, and true/false questions. At the end of the examination when the user clicks a button the total marks have to be displayed in the form of the message.

17) Write a java program that creates menu which appears similar to the menu of notepad application of the Microsoft windows or any editor of your choice.

18) Write a java program that creates dialog box which is similar to the save dialog box of the Microsoft windows or any word processor of your choice.

19) Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication

20) Write a java program to find and replace pattern in a given file.

21) Use inheritance to create an exception super class called ExceptionA and exception sub classes ExceptionB and ExceptionC, where ExceptionB inherits from ExceptionA and ExceptionC inherits from ExceptionB. Write a java program to demonstrate that the catch block for type ExceptionA catches exception of type ExceptionB and ExceptionC.

22) Write a Java program which opens a connection to standard port on well known server, sends the data using socket and prints the returned data.

23) Write a Java program to create a URLConnection and use it to examine the documents properties and content.

24) Write a Java program which uses TCP/IP and Datagrams to communicate client and server.

25) Create an interface for stack with push and pop operations. Implement the stack in two ways: fixed size stack and Dynamic stack (stack size is increased when stack is full).

26) Create multiple threads to access the contents of a stack. Synchronize thread to prevent simultaneous access to push and pop operations.

References:

1. "Java: How to Program", P.J.Deitel and H.M.Deitel, PHI.
2. "Object Oriented Programming through Java", P.Radha Krishna, Universities Press.
3. "Thinking in Java", Bruce Eckel, Pearson Education
4. "Programming in Java", S.Malhotra and S.Choudhary, Oxford Univ. Press.