SEMESTER IV

(This page is intentionally left blank)

Java Programming

B. Sc. (Information Technology)		Semester – IV	
Course Name: Java Programming		**Course Code: USIT401**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objectives:

Upon completion of this course, students will be able to:

- Understand the concept of OOP as well as the purpose and usage principles of inheritance, polymorphism, encapsulation and method overloading.
- Identify classes, objects, members of a class and the relationships among them needed for a specific problem.
- Create Java application programs using sound OOP practices (e.g., interfaces and APIs) and proper program structuring (e.g., by using access control identifies, automatic documentation through comments, error exception handling).
- Use testing and debugging tools to automatically discover errors of Java programs as well as use versioning tools for collaborative programming/editing.
- Develop programs using the Java Collection API as well as the Java standard class library.
- Apply object-oriented programming concepts in problem solving through JAVA.

Unit	Details	Lectures
I	**Introduction:** History, Features of Java, Java Development Kit, Java Application Programming Interface, Java Virtual Machine, Java Program Structure. **Classes:** The Class Object and Its Attributes, Class Methods, Accessing A Method, Method Overloading, Instantiating Objects from A Class, Constructors, this keyword, super keyword, Types of Classes, Scope Rules, Access Modifier, constants, static members of a class, garbage collection.	**12**
II	**Inheritance:** Derived Class Objects, Inheritance and Access Control, Default Base Class Constructors, this and super keywords. Abstract Classes and Interfaces, Abstract Classes, Abstract Methods, **Interfaces**: What Is an Interface? How Is an Interface Different from An Abstract Class? Multiple Inheritance, Defining an Interface, Implementing Interfaces.	**12**
III	**Exceptions:** Catching Java Exceptions, Catching Run-Time Exceptions, Handling Multiple Exceptions, The finally Clause, The throws Clause, Built-in Exceptions in java **Multithreading:** Thread Creations, Thread Life Cycle, Life Cycle Methods, Synchronization, wait() notify() notify all() methods **Packages:** Introduction to predefined packages, User Defined Packages, Access specifier, Java Built-in packages, Array Class, String Class	**12**

IV	**Introduction to JFC and Swing-** Features of the Java Foundation Classes, Swing API Components, JComponent Class, Containers and Panels, Labels, Buttons, RadioButton, Check Boxes, Text-Entry Components, Menus **Layouts:** Flow Layout, Grid Layout, Border Layout **Event Handling:** Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes.	**12**
V	**Advanced Swing Controls:** JScrollPane, Lists and Combo Boxes, Colors and File Choosers, Tables and Trees, JTabbedPane**.** **JDBC**: Introduction, JDBC Architecture, JDBC Drivers, java.sql package, Using Statement, PreparedStatement, CallableStatement, ResultSet	**12**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Core Java 8 for Beginners	Vaishali Shah, Sharnam Shah	SPD	1st	2015
2.	Java: The Complete Reference	Herbert Schildt	McGraw Hill	9th	2014
3.	Murach's beginning Java with Net Beans	Joel Murach , Michael Urban	SPD	1st	2016
4.	Core Java, Volume I: Fundamentals	Hortsman	Pearson	9th	2013
5.	Core Java, Volume II: Advanced Features	Gary Cornell and Hortsman	Pearson	8th	2008
6.	Core Java: An Integrated Approach	R. Nageswara Rao	DreamTech	1st	2008

Course Outcome:

After completing the course, the learner will be able to:

CO1: Learn the architecture of Java

CO2: Identify data types, control flow, classes, inheritance, exceptions and event handling

CO3: Use object-oriented concepts for problem solving real-life applications

CO4: Build GUI programs

CO5 : Create event driven programs using java.

Introduction to Embedded Systems

B. Sc. (Information Technology)		Semester – IV	
Course Name: Introduction to Embedded Systems		**Course Code: USIT402**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objectives:

- To introduce the Building Blocks of Embedded System
- To Educate in Various microcontrollers used in Embedded Development
- To Introduce Bus Communication in processors, Input/output interfacing.
- To impart knowledge in sensors and actuators.
- To familiar with the real world application development using embedded system.

Unit	Details	Lectures
I	**PIC MICROCONTROLLER:** Architecture – memory organization – addressing modes – instruction set – PIC programming in Assembly & C –I/O port, Data Conversion, RAM & ROM Allocation, Timer programming **Advanced ARM Controllers:** Introduction to ARM and its Features, Architecture – memory organization – addressing modes –The ARM Programmer's model -Registers – Pipeline - Interrupts – Coprocessors – Interrupt Structure	**12**
II	**Communication Protocol & Implementation:** Introduction to Communication Protocol, I2C - Interfacing with micro controller using bit-banking method, I2C devices – RTC, Memory, ADC-DAC, Port Expander, SPI (Serial Peripheral Interface), Bluetooth, Wi-Fi and RFID. Understanding Serial, Communication, Bluetooth Communication, SPI Interface ZigBee, Wi-Fi, I^2C, Infrared, RFID, GSM, GPS, PDH/SDH/Ethernet	**12**
III	**Getting Started with Arduino:** Introduction, Arduino Variants, Install the Drivers, Arduino IDE **Basic Functions:** Overview, Structure, Digital I/O Functions, Analog I/O Functions, Advanced I/O Functions, Timer Functions, Communication Functions, Interrupt Functions, Math Functions, Programming Language Reference	**12**

IV	**Using Sensors with the Arduino:** Light Sensitive Sensors, Temperature Sensors, Temperature and Humidity Sensor, Line-Tracking Sensor, Ultrasonic Sensors, Digital Infrared Motion Sensor, Joystick Module, Gas Sensor, Hall Sensor, Color Sensor, Digital Tilt Sensor, Triple Axis Acceleration Sensor, Analog Sound Sensor, Voice Recognition Module, Digital Vibration Sensor, Flame Sensor, Capacitive Touch Sensor **Electromechanical Control Using the Arduino:** DC Motor, Stepper Motor, Servo Motor	**12**
V	**Wireless Control Using the Arduino:** Infrared Transmitter and Receiver, Wireless Radio Frequency, Bluetooth, GSM/GPRS, Wi-Fi **Case Studies:** • Air Quality Monitor Using Arduino • A Fire-Fighting Robot Using Arduino • Intelligent Lock System Using Arduino	**12**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Programming Embedded Systems in C and C++	Michael Barr	O'Reilly	First	1999
2.	Introduction to embedded systems	Shibu K V	Tata Mcgraw-Hill	First	2012
3.	The 8051 Microcontroller and Embedded Systems	Muhammad Ali Mazidi	Pearson	Second	2011
4.	Embedded Systems	Rajkamal	Tata Mcgraw-Hill		

Course Outcome:

CO1: Differentiate between general purpose and embedded systems

CO2: Discuss the characteristics and quality attributes of embedded systems

CO3: Use different types of sensors for appropriately

CO4: Design and develop embedded systems

Computer Oriented Statistical Techniques

B. Sc. (Information Technology)		Semester – IV	
Course Name: Computer Oriented Statistical Techniques		**Course Code: USIT403**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objectives:

1. To learn the different methods of calculating the central tendencies.
2. To introduce the moments, skewness and kurtosis.
3. To learn scientific view to conduct the survey in proper way to collect the data about specific perspective.
4. To Learn variety of probability sampling methods for selecting a sample from a population.
5. To learn the sampling theory and testing of hypothesis and making inferences.
6. To introduce the students with understanding of the curve fitting, regression and correlation techniques.

Unit	Details	Lectures
I	**The Mean, Median, Mode, and Other Measures of Central Tendency**: Index, or Subscript, Notation, Summation Notation, Averages, or Measures of Central Tendency ,The Arithmetic Mean , The Weighted Arithmetic Mean ,Properties of the Arithmetic Mean ,The Arithmetic Mean Computed from Grouped Data ,The Median ,The Mode, The Empirical Relation Between the Mean, Median, and Mode, The Geometric Mean G, The Harmonic Mean H ,The Relation Between the Arithmetic, Geometric, and Harmonic Means, The Root Mean Square, Quartiles, Deciles, and Percentiles, Software and Measures of Central Tendency. **The Standard Deviation and Other Measures of Dispersion**: Dispersion, or Variation, The Range, The Mean Deviation, The Semi-Interquartile Range, The 10–90 Percentile Range, The Standard Deviation, The Variance, Short Methods for Computing the Standard Deviation, Properties of the Standard Deviation, Charlie's Check, Sheppard's Correction for Variance, Empirical Relations Between Measures of Dispersion, Absolute and Relative Dispersion; Coefficient of Variation, Standardized Variable; Standard Scores, Software and Measures of Dispersion. **Introduction to R:** Basic syntax, data types, variables, operators, control statements, R-functions, R –Vectors, R – lists, R Arrays.	**12**

II	**Moments, Skewness, and Kurtosis** : Moments , Moments for Grouped Data ,Relations Between Moments , Computation of Moments for Grouped Data, Charlie's Check and Sheppard's Corrections, Moments in Dimensionless Form, Skewness, Kurtosis, Population Moments, Skewness, and Kurtosis, Software Computation of Skewness and Kurtosis. **Elementary Probability Theory**: Definitions of Probability, Conditional Probability; Independent and Dependent Events, Mutually Exclusive Events, Probability Distributions, Mathematical Expectation, Relation Between Population, Sample Mean, and Variance, Combinatorial Analysis, Combinations, Stirling's Approximation to n!, Relation of Probability to Point Set Theory, Euler or Venn Diagrams and Probability. **Elementary Sampling Theory** : Sampling Theory, Random Samples and Random Numbers, Sampling With and Without Replacement, Sampling Distributions, Sampling Distribution of Means, Sampling Distribution of Proportions, Sampling Distributions of Differences and Sums, Standard Errors, Software Demonstration of Elementary Sampling Theory.	**12**
III	**Statistical Estimation Theory**: Estimation of Parameters, Unbiased Estimates, Efficient Estimates, Point Estimates and Interval Estimates; Their Reliability, Confidence-Interval Estimates of Population Parameters, Probable Error. **Statistical Decision Theory:** Statistical Decisions, Statistical Hypotheses, Tests of Hypotheses and Significance, or Decision Rules, Type I and Type II Errors, Level of Significance, Tests Involving Normal Distributions, Two-Tailed and One-Tailed Tests, Special Tests, Operating-Characteristic Curves; the Power of a Test, p-Values for Hypotheses Tests, Control Charts, Tests Involving Sample Differences, Tests Involving Binomial Distributions. **Statistics in R:** mean, median, mode, Normal Distribution , Binomial Distribution, Frequency Distribution in R.	**12**
IV	**Small Sampling Theory**: Small Samples, Student's t Distribution, Confidence Intervals, Tests of Hypotheses and Significance, The Chi-Square Distribution, Confidence Intervals for Sigma , Degrees of Freedom, The F Distribution. **The Chi-Square Test**: Observed and Theoretical Frequencies, Definition of chi-square, Significance Tests, The Chi-Square Test for Goodness of Fit, Contingency Tables, Yates' Correction for Continuity, Simple Formulas for Computing chi-square, Coefficient of Contingency, Correlation of Attributes, Additive Property of chi-square.	**12**
V	**Curve Fitting and the Method of Least Squares:** Relationship Between Variables, Curve Fitting, Equations of Approximating Curves, Freehand Method of Curve Fitting, The Straight Line, The Method of Least Squares, The Least-Squares Line, Nonlinear Relationships, The	**12**

	Least-Squares Parabola, Regression, Applications to Time Series, Problems Involving More Than Two Variables. **Correlation Theory:** Correlation and Regression, Linear Correlation, Measures of Correlation, The Least-Squares Regression Lines, Standard Error of Estimate, Explained and Unexplained Variation, Coefficient of Correlation, Remarks Concerning the Correlation Coefficient, Product-Moment Formula for the Linear Correlation Coefficient, Short Computational Formulas, Regression Lines and the Linear Correlation Coefficient, Correlation of Time Series, Correlation of Attributes, Sampling Theory of Correlation, Sampling Theory of Regression.	

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	STATISTICS	Murray R. Spiegel, Larry J. Stephens.	McGRAW – HILL ITERNATIONAL	FOURTH	
2.	A Practical Approach using R	R.B. Patil, H.J. Dand and R. Bhavsar	SPD	1st	2017
3.	FUNDAMENTAL OF MATHEMATICAL STATISTICS	S.C. GUPTA and V.K. KAPOOR	SULTAN CHAND and SONS	ELEVENTH REVISED	2011
4.	MATHEMATICAL STATISTICS	J.N. KAPUR and H.C. SAXENA	S. CHAND	TWENTIETH REVISED	2005

Course Outcome: Upon the successful completion of the course, students will be able to:

CO 1: To calculate and apply measures of central tendencies and measures of dispersion -- grouped and ungrouped data cases.
CO 2: To calculate the moments, skewness and kurtosis by various methods.
CO 3: How to apply discrete and continuous probability distributions to various business problems.
CO 4: Perform Test of Hypothesis as well as calculate confidence interval for a population parameter for single sample and two sample cases. Understand the concept of p-values
CO 5: Apply simple linear regression and correlation model to real life examples.

Software Engineering

B. Sc. (Information Technology)		Semester – IV	
Course Name: Software Engineering		**Course Code: USIT404**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objective:

- Develop the software projects or prototypes by understanding the requirements.
- Meet the project deadlines along with the number of resources and type of tasks to be carried out.
- Evaluate and analyze the SDLC and basic architecture SRS documents.
- Help to understand the software design and coding techniques.
- Understand the software testing principles.
- Understand the concept project management.
- Identify various concepts of Advanced UML techniques

Unit	Details	Lectures
I	**Introduction:** What is software engineering? Software Development Life Cycle, Requirements Analysis, Software Design, Coding, Testing, Maintenance etc. **Software Requirements:** Functional and Non-functional requirements, User Requirements, System Requirements, Interface Specification, Documentation of the software requirements. **Software Processes:** Process and Project, Component Software Processes. **Software Development Process Models.** • Waterfall Model. • Prototyping. • Iterative Development. • Rational Unified Process. • The RAD Model • Time boxing Model. **Agile software development**: Agile methods, Plan-driven and agile development, Extreme programming, Agile project management, Scaling agile methods.	**12**
II	**Socio-technical system:** Essential characteristics of socio technical systems, Emergent System Properties, Systems Engineering, Components of system such as organization, people and computers, Dealing Legacy Systems. **Critical system:** Types of critical system, A simple safety critical system, Dependability of a system, Availability and Reliability, Safety and Security of Software systems.	**12**

	Requirements Engineering Processes: Feasibility study, Requirements elicitation and analysis, Requirements Validations, Requirements Management. **System Models:** Models and its types, Context Models, Behavioural Models, Data Models, Object Models, Structured Methods.	
III	**Architectural Design:** Architectural Design Decisions, System Organisation, Modular Decomposition Styles, Control Styles, Reference Architectures. **User Interface Design:** Need of UI design, Design issues, The UI design Process, User analysis, User Interface Prototyping, Interface Evaluation. **Project Management** Software Project Management, Management activities, Project Planning, Project Scheduling, Risk Management. **Quality Management:** Process and Product Quality, Quality assurance and Standards, Quality Planning, Quality Control, Software Measurement and Metrics.	**12**
IV	**Verification and Validation:** Planning Verification and Validation, Software Inspections, Automated Static Analysis, Verification and Formal Methods. **Software Testing:** System Testing, Component Testing, Test Case Design, Test Automation. **Software Measurement**: Size-Oriented Metrics, Function-Oriented Metrics, Extended Function Point Metrics **Software Cost Estimation:** Software Productivity, Estimation Techniques, Algorithmic Cost Modelling, Project Duration and Staffing	**12**
V	**Process Improvement:** Process and product quality, Process Classification, Process Measurement, Process Analysis and Modeling, Process Change, The CMMI Process Improvement Framework. **Service Oriented Software Engineering:** Services as reusable components, Service Engineering, Software Development with Services. **Software reuse:** The reuse landscape, Application frameworks, Software product lines, COTS product reuse. **Distributed software engineering**: Distributed systems issues, Client–server computing, Architectural patterns for distributed systems, Software as a service	**12**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Software Engineering, edition,	Ian Somerville	Pearson Education.	Ninth	
2.	Software Engineering	Pankaj Jalote	Narosa Publication		

3.	Software engineering, a practitioner's approach	Roger Pressman	Tata Mcgraw-hill	Seventh	
4.	Software Engineering principles and practice	WS Jawadekar	Tata Mcgraw-hill		
5.	Software Engineering-A Concise Study	S.A Kelkar	PHI India.		
6.	Software Engineering Concept and Applications	Subhajit Datta	Oxford Higher Education		
7.	Software Design	D.Budgen	Pearson education	2nd	
8.	Software Engineering	KL James	PHI	EEE	2009

Course Outcome:

After completing the course, the learner will be able to:

CO1: Understand software engineering

CO2: Apply software engineering principles

CO3: Discuss various approaches to verification and validation of software including testing, measurements and estimation of software products

CO4: Create software using different software development models

Computer Graphics and Animation

B. Sc. (Information Technology)		Semester – IV	
Course Name: Computer Graphics and Animation		**Course Code: USIT405**	
Periods per week (1 Period is 50 minutes)		**5**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Theory Examination**	**2½**	**75**
	Internal	**--**	**25**

Course Objectives:

1. To train the students to acquire skills in generating marketable computer graphics and animated pictures, especially in the area of advertisements.
2. To train the students to acquire skills and mastery in the use of different software producing graphics and animation.
3. The course introduces the basic concepts of computer graphics.
4. It provides the necessary theoretical background and demonstrates the application of computer science to graphics.
5. The course further allows students to develop programming skills in computer graphics through programming assignments.

Unit	Details	Lectures
I	**Introduction to Computer Graphics:** Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Input Devices for Operator Interaction, Active and Passive Graphics Devices, Display Technologies, Storage Tube Graphics Displays, Calligraphic Refresh Graphics Displays, Raster Refresh (Raster-Scan) Graphics Displays, Cathode Ray Tube Basics, Color CRT Raster Scan Basics, Video Basics, The Video Controller, Random-Scan Display Processor, LCD displays. **Scan conversion** – Digital Differential Analyzer (DDA) algorithm, Bresenhams' Line drawing algorithm. Bresenhams' method of Circle drawing, Midpoint Circle Algorithm, Midpoint Ellipse Algorithm, Mid-point criteria, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Clipping Lines algorithms– Cyrus-Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components.	**12**
II	**Two-Dimensional Transformations:** Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The Unit Square, Solid Body Transformations, Rotation About an Arbitrary Point, Reflection	**12**

	through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations. **Three-Dimensional Transformations:** Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three-Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations, Affine and Perspective Geometry, Perspective Transformations, Techniques for Generating Perspective Views, Vanishing Points, the Perspective Geometry and camera models, Orthographic Projections, Axonometric Projections, Oblique Projections, View volumes for projections.	
III	**Viewing in 3D** Stages in 3D viewing, Canonical View Volume (CVV), Specifying an Arbitrary 3D View, Examples of 3D Viewing, The Mathematics of Planar Geometric Projections, Combined transformation matrices for projections and viewing, Coordinate Systems and matrices, camera model and viewing pyramid. **Light:** Radiometry, Transport, Equation, Photometry **Color:** Colorimetry, Color Spaces, Chromatic Adaptation, Color Appearance	**12**
IV	**Visible-Surface Determination:** Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, The z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Area sub-division method, BSP trees, Visible-Surface Ray Tracing, comparison of the methods. **Plane Curves and Surfaces:** Curve Representation, Nonparametric Curves, Parametric Curves, Parametric Representation of a Circle, Parametric Representation of an Ellipse, Parametric Representation of a Parabola, Parametric Representation of a Hyperbola, Representation of Space Curves, Cubic Splines, , Bezier Curves, B-spline Curves, B-spline Curve Fit, B-spline Curve Subdivision, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces.	**12**
V	**Computer Animation:** Principles of Animation, Key framing, Deformations, Character Animation, Physics-Based Animation, Procedural Techniques, Groups of Objects. **Image Manipulation and Storage:** What is an Image? Digital image file formats, Image compression standard – JPEG, Image Processing - Digital image enhancement, contrast stretching, Histogram Equalization, smoothing and median Filtering.	**12**

Books and References:					
Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	Computer Graphics - Principles and Practice	J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes	Pearson	2nd	
2.	Steve Marschner, Peter Shirley	Fundamentals of Computer Graphics	CRC press	4th	2016
3.	Computer Graphics	Hearn, Baker	Pearson	2nd	
4.	Principles of Interactive Computer Graphics	William M. Newman and Robert F. Sproull	TMH	2nd	
5.	Mathematical Elements for CG	D. F. Rogers, J. A. Adams	TMH	2nd	

After completion of the course students are supposed to be able to:

CO 1. Understand the basics of computer graphics, different graphics systems and applications of computer graphics
CO 2. Compare various algorithms for scan conversion and filling of basic objects
CO 3. Use of geometric transformations on graphics objects and their application in composite form.
CO 4. Extract scene with different clipping methods and its transformation to graphics display device.
CO 5. Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
CO 6. Render projected objects to naturalize the scene in 2D view and use of illumination models
CO 7. Understand the core concepts and mathematical foundations of computer graphics
CO 8. Know the fundamental computer graphics algorithms and data structures
CO 9. Understand an overview of different modeling approaches and methods
CO 10. Apply basic shading and texture mapping techniques
CO 11. Understand light interaction with 3D scenes
CO 12. Explain the applications, areas, and graphic pipeline, display and hardcopy technologies.
CO 13. Apply and compare the algorithms for drawing 2D images also explain aliasing, anti-aliasing and half toning techniques.
CO 14. Discuss OpenGL application programming Interface and apply it for 2D & 3D computer graphics.
CO 15. Analyze and apply clipping algorithms and transformation on 2D images.
CO 16. Solve the problems on viewing transformations and explain the projection and hidden surface removal algorithms.
CO 17. Apply basic ray tracing algorithm, shading, shadows, curves and surfaces and also solve the problems of curves.

Java Programming Practical

B. Sc. (Information Technology)		Semester – III	
Course Name: Java Programming Practical		**Course Code: USIT4P----**	
Periods per week (1 Period is 50 minutes)		**3**	
Credits		**2**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**
	Internal	**--**	

List of Practical:	
1.	**OOPs concepts in Java – 1**
a.	Write a program to create a class and implement a default, overloaded and copy Constructor.
b.	Write a program to create a class and implement the concepts of Method Overloading
c.	Write a program to create a class and implement the concepts of Static methods
2.	**OOPs concepts in Java – 2**
a.	Write a program to implement the concepts of Inheritance and Method overriding
b.	Write a program to implement the concepts of Abstract classes and methods
c.	Write a program to implement the concept of interfaces
3.	**Exceptions**
a.	Write a program to raise built-in exceptions and raise them as per the requirements
b.	Write a program to define user defined exceptions and raise them as per the requirements
4.	**Multithreading:** Write a java application to demonstrate 5 bouncing balls of different colors using threads.
5.	**JDBC**
a.	Write a JDBC program that displays the data of a given table in a GUI Table.
b.	Write a JDBC program to Show the details of a specified product from a given table selected using Combobox.
c.	Write a GUI application to Navigate forward and reverse result set data.
6.	**Swing**
a.	Create a swing application that randomly changes color on button click.
b.	Create a Swing application to demonstrate use of TextArea using scrollpane to show contest of text file in textarea selected using file chooser.
c.	Create a Swing application to demonstrate use of scrollpane to change its color selected using colour chooser.
7.	**Layouts: Write programs for the following layouts:**

a.	Flow Layout
b.	Grid Layout
c.	Border Layout
8.	**Events: Write programs to demonstrate the following events:**
a.	ActionEvent
b.	MouseEvent
c.	KeyEvent
d.	SelectionEvent
e.	FocusEvent
9.	**Demonstrate the use of Adapter Class in Event Handling**
10.	**Demonstrate the use of Anonymous Inner Class in Event Handling**

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Core Java 8 for Beginners	Vaishali Shah, Sharnam Shah	SPD	1st	2015
2.	Java: The Complete Reference	Herbert Schildt	McGraw Hill	9th	2014
3.	Murach's beginning Java with Net Beans	Joel Murach , Michael Urban	SPD	1st	2016
4.	Core Java, Volume I: Fundamentals	Hortsman	Pearson	9th	2013
5.	Core Java, Volume II: Advanced Features	Gary Cornell and Hortsman	Pearson	8th	2008
6.	Core Java: An Integrated Approach	R. Nageswara Rao	DreamTech	1st	2008

Introduction to Embedded Systems Practical

B. Sc. (Information Technology)		**Semester – IV**	
Course Name: Introduction to Embedded Systems Practical		**Course Code: USIT4P2**	
Periods per week **1 Period is 50 minutes**	**Lectures per week**	**3**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**

List of Practical: All practicals to be done online using TinkerCAD	
1.	**Introduction to Arduino**
	Introduction to Arduino circuits and breadboarding
	Blinking of LEDs
2.	Program using Light Sensitive Sensors
3.	Program using temperature sensors
4.	Programs using humidity sensors
5.	Programs using Line tracking sensors
6.	Programs using Ultrasonic Sensors
7.	Programs using digital infrared motion sensors
8.	Programs using gas sensors
9.	Programs using servo motors
10.	Programs making Joystick with Arduino

Computer Oriented Statistical Techniques Practical

B. Sc. (Information Technology)		**Semester – IV**	
Course Name: Computer Oriented Statistical Techniques Practical		**Course Code: USIT4P3**	
Periods per week **1 Period is 50 minutes**	**Lectures per week**	**3**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**

List of Practical	
1.	Using R/Python execute the basic commands, array, list and frames.
2.	Create a Matrix using R/Python and Perform the operations addition, inverse, transpose and multiplication operations.
3.	Using R/Python Execute the statistical functions: mean, median, mode, quartiles, range, inter quartile range histogram
4.	Using R/Python import the data from Excel / .CSV file and Perform the above functions.
5.	Using R/Python import the data from Excel / .CSV file and Calculate the standard deviation, variance, co-variance.
6.	Using R/Python import the data from Excel / .CSV file and draw the skewness.
7.	Import the data from Excel / .CSV and perform the hypothesis testing.
8.	Import the data from Excel / .CSV and perform the Chi-squared Test.
9.	Using R/Python perform the binomial and normal distribution on the data.
10.	a. Perform the Linear Regression using R/Python.
	b. Compute the Least squares means using R/Python.
	c. Compute the Linear Least Square Regression using R/Python

Books and References:

Sr. No.	**Title**	**Author/s**	**Publisher**	**Edition**	**Year**
1.	A Practical Approach to R Tool	R.B. Patil, H.J. Dand and R. Dahake	SPD	First	2011
2.	STATISTICS	Murray R. Spiegel, Larry J. Stephens.	McGRAW –HILL INTERNATIONAL	FOURTH	2006

Software Engineering Practical

B. Sc. (Information Technology)		Semester – IV	
Course Name: Software Engineering Practical		**Course Code: USIT4P4**	
Periods per week **1 Period is 50 minutes**	**Lectures per week**	**3**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**

List of Practical (To be executed using Star UML or any similar software)	
1.	Study and implementation of class diagrams.
2.	Study and implementation of Use Case Diagrams.
3.	Study and implementation of Entity Relationship Diagrams.
4.	Study and implementation of Sequence Diagrams.
5.	Study and implementation of State Transition Diagrams.
6.	Study and implementation of Data Flow Diagrams.
7.	Study and implementation of Collaboration Diagrams.
8.	Study and implementation of Activity Diagrams.
9.	Study and implementation of Component Diagrams.
10.	Study and implementation of Deployment Diagrams.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
3.	Object - Oriented Modeling and Design	Michael Blaha, James Rumbaugh	Pearson		2011
4.	Learning UML 2. 0	Kim Hamilton, Russ Miles	O'Reilly Media		2006
5.	The unified modeling language user guide	Grady Booch, James Rumbaugh, Ivar Jacobson	Addison-Wesley		2005
6.	UML A Beginners Guide	Jason T. Roff	McGraw Hill Professional		2003

Computer Graphics and Animation

B. Sc. (Information Technology)		Semester – IV	
Course Name: Computer Graphics and Animation		**Course Code: USIT4P5**	
Periods per week **1 Period is 50 minutes**	**Lectures per week**	**3**	
		Hours	**Marks**
Evaluation System	**Practical Examination**	**2½**	**50**

List of Practical	
1.	**Solve the following:**
a.	Study and enlist the basic functions used for graphics in C / C++ / Python language. Give an example for each of them.
b.	Draw a co-ordinate axis at the center of the screen.
2.	**Solve the following:**
a.	Divide your screen into four region, draw circle, rectangle, ellipse and half ellipse in each region with appropriate message.
b.	Draw a simple hut on the screen.
3.	**Draw the following basic shapes in the center of the screen :**
	i. Circle ii. Rectangle iii. Square iv. Concentric Circles v. Ellipse vi. Line
4.	**Solve the following:**
a.	Develop the program for DDA Line drawing algorithm.
b.	Develop the program for Bresenham's Line drawing algorithm.
5.	**Solve the following:**
a.	Develop the program for the mid-point circle drawing algorithm.
b.	Develop the program for the mid-point ellipse drawing algorithm.
6.	**Solve the following:**
a.	Write a program to implement 2D scaling.
b.	Write a program to perform 2D translation
7.	**Solve the following:**
a.	Perform 2D Rotation on a given object.
b.	Program to create a house like figure and perform the following operations. i. Scaling about the origin followed by translation. ii. Scaling with reference to an arbitrary point. iii. Reflect about the line y = mx + c.

8.	**Solve the following:**
a.	Write a program to implement Cohen-Sutherland clipping.
b.	Write a program to implement Liang - Barsky Line Clipping Algorithm
9.	**Solve the following:**
a.	Write a program to fill a circle using Flood Fill Algorithm.
b.	Write a program to fill a circle using Boundary Fill Algorithm.
10.	**Solve the following:**
a.	Develop a simple text screen saver using graphics functions.
b.	Perform smiling face animation using graphic functions.
c.	Draw the moving car on the screen.

Books and References:

Sr. No.	Title	Author/s	Publisher	Edition	Year
1.	Computer Graphics - Principles and Practice	J. D. Foley, A. Van Dam, S. K. Feiner and J. F. Hughes	Pearson Education	Second Edition	
2.	Steve Marschner, Peter Shirley	Fundamentals of Computer Graphics	CRC press	Fourth Edition	2016
3.	Computer Graphics	Hearn, Baker	Pearson Education	Second	
4.	Principles of Interactive Computer Graphics	William M. Newman and Robert F. Sproull	Tata McGraw Hill	Second	

Evaluation Scheme:

1. **Internal Evaluation (25 Marks).**
   - **i. Test: 1 Class test of 20 marks. (Can be taken online)**

Q	Attempt ***any four*** of the following:	20
a.		
b.		
c.		
d.		
e.		
f.		

   - **ii. 5 marks: Active participation in the class, overall conduct, attendance.**

2. **External Examination: (75 marks)**

	All questions are compulsory	
Q1	(Based on Unit 1) Attempt ***any three*** of the following:	15
a.		
b.		
c.		
d.		
e.		
f.		
Q2	(Based on Unit 2) Attempt ***any three*** of the following:	15
Q3	(Based on Unit 3) Attempt ***any three*** of the following:	15
Q4	(Based on Unit 4) Attempt ***any three*** of the following:	15
Q5	(Based on Unit 5) Attempt ***any three*** of the following:	15

3. **Practical Exam: 50 marks**

A Certified copy journal is essential to appear for the practical examination.

1.	Practical Question 1	20
2.	Practical Question 2	20
3.	Journal	5
4.	Viva Voce	5

OR

1.	Practical Question	40
2.	Journal	5
3.	Viva Voce	5