Government of Karnataka
Department of Technical Education
Bengaluru

	Course Title: **OOP With Java**		
	Scheme (L:T:P) : **4:0:0**	Total Contact Hours: **52**	Course Code: **15CS42T**
	Type of Course: **Lectures, Self Study & Student Activity.**	Credit **:04**	Core/ Elective: **Core**
CIE- 25 Marks			SEE- 100 Marks

Prerequisites

Knowledge of programming concepts.

Course Objectives

To learn and implement object-oriented features such as encapsulation, inheritance and polymorphism along with error-handling techniques using Java.

Course Outcome

On successful completion of the course, the students will be able to attain below Course Outcome (CO):

Course outcome		CL	Linked PO	Teaching Hours
CO1	Discuss the OOP's concept and Apply the concepts to design, implement, compile, test and execute simple Java programs.	***U, A***	**2,3,4,8,10**	**8**
CO2	Explain the concepts related to classes and Use built-in methods of String and String Buffer classes.	***U, A***	**2,3,4,8,10**	**14**
CO3	Define Inheritance and Discover Interface with programs	***U, A***	**2,3,4,8,10**	**6**
CO4	Illustrate Packages and articulate with simple programs	***U, A***	**2,3,4,8,10**	**8**
CO5	Illustrate multithreading concepts by experimenting with programs	***U, A***	**2,3,4,8,10**	**8**
CO6	Interpret different types of Exceptions by solving programs.	***U, A***	**2,3,4,8,10**	**8**
		Total sessions		**52**

Legends: R = Remember U= Understand; A= Apply and above levels (Bloom's revised taxonomy)

Course-PO Attainment Matrix

Course	Programme Outcomes									
	1	2	3	4	5	6	7	8	9	10
OOP With Java	**-**	**3**	**3**	**3**	**-**	**-**	**-**	**3**	**-**	**3**

Level 3- Highly Addressed, Level 2-Moderately Addressed, Level 1-Low Addressed.

Method is to relate the level of PO with the number of hours devoted to the COs which address the given PO.

If ≥40% of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 3

If 25 to 40% of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 2

If 5 to 25% of classroom sessions addressing a particular PO, it is considered that PO is addressed at Level 1

If < 5% of classroom sessions addressing a particular PO, it is considered that PO is considered not-addressed.

Course Content and Blue Print of Marks for SEE

Unit No	Unit Name	Hour	Questions to be set for SEE			Marks Weightage	Marks Weightage (%)
			R	U	A	A	
I	Introduction of Java	**8**	5	10	7	22	15.38
II	Classes, Objects and Methods; Strings and String Buffer Classes	**14**	-	18	10	38	27.00
III	Interface: Multiple Inheritance	**6**	-	6	10	16	11.53
IV	Packages: Putting Classes Together	**8**	-	13	10	23	15.38
V	Multithreaded Programming	**8**	-	13	10	23	15.38
VI	Managing Errors and Exceptions	**8**	-	13	10	23	15.38
	Total	**52**	**5**	**73**	**57**	**145**	**100**

UNIT I: Introduction of Java 06 Hrs

Fundamentals of Object Oriented Programming- Introduction, Object oriented Paradigm, Basic Concepts of OOP, Benefits of OOP, Applications of OOP.

Java Evolution - Java history, Java Features, How Java Differs from C and C++, Java and World Wide Web, Java Environment, Simple Java Program, An Application with Two Classes, Java Program Structure, Java Tokens, Java Statements, Implementing a Java Program, Java Virtual Machine, Command Line Arguments, Programming Style, Constants, Variables, Data Types, Scope of Variables, Symbolic Constants, Type Casting, Standard Default Values, Special Operators, Mathematical Functions, Labelled Loops (break & Continue) Operators and Expressions, Decision Making, Branching & Looping.

UNIT II: Classes, Objects and Methods; Strings and String Buffer Classes 14Hrs

Classes, Objects and Methods - Introduction, Defining a Class, Fields Declaration, Methods Declaration, Creating Objects, Accessing Class Members, Constructors, Methods Overloading, Static Members, Nesting of Methods, Inheritance: Extending a Class, Overriding Methods, Final Variables and Methods, Final Classes, Finalize Methods, Abstract Methods and Classes, Methods with Variable arguments (Varargs),Visibility Control.
Strings and String Buffer Classes - Strings, Vectors, Wrapper classes, Enumerated Types, Annotations.

UNIT III: Interface: Multiple Inheritance 10Hrs

Introduction, Defining Interfaces, Extending Interfaces, Implementing Interfaces, Accessing Interface Variables.

UNIT IV: Packages: Putting Classes Together 08Hrs

Introduction, Java API Packages, Using System Packages, Naming Conventions, Creating Packages, Accessing a Package, Using a Package, Adding a Class to a Package, Hiding Classes, Static Import.

UNIT V: Multithreaded Programming 06 Hrs

Introduction, Creating Threads, Extending the Thread Class, Stopping and Blocking a Thread, Life Cycle of a Thread, Using Thread Methods, Thread Exceptions, Thread Priority, Synchronization, Implementing the 'Runnable' Interface, Inter-thread Communication.

UNIT VI: Managing Errors and Exceptions 08 Hrs

Introduction, Types of Errors, Exceptions, Syntax of Exception Handling Code, Multiple Catch Statements, Using Finally Statement, Throwing Our Own Exceptions.

Text books

Programming with Java, 6th edition, Balagurusamy, Mc Graw Hill, ISBN 13- 9789351343202 ISB 10- 9351343200

References

1. Complete Reference Java J2se, Herbert Schildt, Tata McGraw Hill, ISBN 9780070598782
2. Java 6 Programming – Black Book Wiley India Pvt ltd
3. Programming in JAVA2 – Dr. K. Somasundaram Jaico Publish
4. Programming in JAVA – S.S. Khandare – S. Chand Publish

E-learning resources

http://www.Javatpoint.com/Java-tutorial
http://www.tutorialspoint.com/Java/
http://www.indiabix.com/technical/core-Java/

Suggested list of student activities

Note: The following activities or similar activities for assessing CIE (IA) for 5 marks (Any one)

Student activity like mini-project, surveys, quizzes, etc. should be done in group of 3-5 students.

1. Each group should do any one of the following type activity or any other similar activity related to the course and before conduction, get it approved from concerned Course coordinator and Program Coordinator
2. Each group should conduct different activity and no repeating should occur.

- Hotel Management System, E-Bill Board, Online insurance, Online Mobile, Contributor, Online Restaurant, Public Distribution System, SECURE E-banking security,

District medical data center, Visit different sites relevant to topics. Listen to the lectures and submit a handwritten report, etc.

Course Delivery

The course will be delivered through lectures and Power point presentations/ Video

Course Assessment and Evaluation Scheme

Method	What		To whom	When/Where (Frequency in the course)	Max Marks	Evidence collected	Course outcomes
Direct Assessment	CIE	IA	Students	Three IA tests (Average of three tests will be computed)	20	Blue books	1 to 6
				Student activities	05	Activity Reports	1 to 6
				Total	**25**		
	SEE	End Exam		**End of the course**	**100**	Answer scripts at BTE	1 to 6
Indirect Assessment	Student Feedback on course		Students	Middle of the course		Feedback forms	1 to 3 Delivery of course
	End of Course Survey			End of the course		Questionnaires	1 to 6 Effectiveness of Delivery of instructions & Assessment Methods

Note: I.A. test shall be conducted for 20 marks. Average marks of three tests shall be rounded off to the next higher digit.

Questions for CIE and SEE will be designed to evaluate the various educational components (Bloom's taxonomy) such as:

Sl. No	Bloom's Category	%
1	Remembrance	10
2	Understanding	30
3	Application	60

Note to IA verifier*: *The following documents to be verified by CIE verifier at the end of semester

1. Blue books (20 marks)
2. Student suggested activities report for 5 marks
3. Student feedback on course regarding Effectiveness of Delivery of instructions & Assessment Methods.

FORMAT OF I A TEST QUESTION PAPER (CIE)

Test/Date and Time	Semester/year	Course/Course Code	Max Marks
Ex: I test/6 th weak of sem 10-11 Am	I/II SEM		20
	Year:		

Name of Course coordinator : Units:__
CO's:____

Question no	Question	MARKS	CL	CO	PO
1					
2					
3					
4					

Note: Internal choice may be given in each CO at the same cognitive level (CL).

MODEL QUESTION PAPER (CIE)

Test/Date and Time	Semester/year	Course/Course Code	Max Marks
Ex: I test/6 th week of sem 10-11 AM	IV SEM	OOP With Java	20
	Year: 2015-16	Course code:15CS42T	

Name of Course coordinator :
Units:1,2 Co: 1,2

Note: Answer all questions

Question no	Question		CL	CO	PO
1	Define the following OOPS concepts a. Inheritance b. Polymorphism	(5)	U	1	1,2
2	Illustrate Java communication with a web page	(5)	A	1	1,2
3	Explain any five string buffer methods	(5)	U	2	1,2
4	Write a Java Programme to illustrate to illustrate vectors.	(5)		2	1,2

Format for Student Activity Assessment

DIMENSION	Unsatisfactory 1	Developing 2	Satisfactory 3	Good 4	Exemplary 5	Score
Collection of data	Does not collect any information relating to the topic	Collects very limited information; some relate to the topic	Collects some basic information; refer to the topic	Collects relevant information; concerned to the topic	Collects a great deal of information; all refer to the topic	3
Fulfill team's roles & duties	Does not perform any duties assigned to the team role	Performs very little duties	Performs nearly all duties	Performs all duties	Performs all duties of assigned team roles with presentation	4
Shares work equally	Always relies on others to do the work	Rarely does the assigned work; often needs reminding	Usually does the assigned work; rarely needs reminding	Does the assigned job without having to be reminded.	Always does the assigned work without having to be reminded and on given time frame	3
Listen to other Team mates	Is always talking; never allows anyone else to speak	Usually does most of the talking; rarely allows others to speak	Listens, but sometimes talk too much	Listens and contributes to the relevant topic	Listens and contributes precisely to the relevant topic and exhibit leadership qualities	3
					TOTAL	**13/4=3.25=4**

All student activities should be done in a group of 4-5 students with a team leader.

Diploma in Computer science & Engineering

IV- Semester

Course Title: OOP With Java Programming

Time: **3 Hours** Max Marks**: 100**

PART-A

Answer any SIX questions. Each carries 5 marks. 5X6=30 Marks

1. List any five major differences between C++ and Java.
2. Distinguish between Inheritance &Polymorphism.
3. Define constructors. List its special properties.
4. Compare arrays and vectors.
5. Explain how to add a class to a package with an example.
6. Explain the various forms of interface implementation.
7. Explain yield(),sleep() and stop() methods of a thread.
8. List the Java exceptions.
9. Illustrate with an example nested try statement.

.

PART-B

Answer any SEVEN full questions each carries 10 marks. 10X7=70 Marks

1. Explain the features of Java.
2. Explain class definition with fields and method declaration.
3. Define static member. Write a program to illustrate static members.
4. Define inheritance. Explain different forms of inheritance.
5. Write a program to implement interfaces
6. Explain Java API packages
7. Write a Package program to demonstrate basic arithmetic operators
8. Explain the life cycle of thread.
9. Explain with an example thread creation by implementing runnable Interface..
10. Write a program for throwing your own exception.

Diploma in Computer Science & Engineering

IV Semester

Course Title: OOP with Java

CO	Question	CL	Marks
I	Give the definition of object oriented programming and how it is different from procedure-oriented programming	U	05
	Explain the organization of data and methods in an object oriented programming.	U	
	List the unique advantages of an object oriented programming	U	
	Give the definition of object and classes. Explain representation of an object with an example.	U	
	Define the following (i) Data abstraction (ii) Data encapsulation	U	
	Define the following OOPS concepts b. Inheritance c. Polymorphism	U	
	Distinguish between objects and classes	U	
	Distinguish between Data abstraction & Data encapsulation	U	
	Distinguish between Inheritance &Polymorphism	U	
	Distinguish between Dynamic binding & message passing.	U	
	Explain Dynamic binding & message passing	U	
	List any five advantages of OOP	A	
	List any five areas of application of OOPS technology	A	
	Java is platform independent language. Justify	U	
	Discuss how Java is more secured than other language	U	
	List any features of Java.	A	
	List any five major differences between C and Java.	U	
	List any five major differences between C++ and Java	U	
	Discuss the contributions of Java to the world wide web.	A	
	Illustrate Java communication with a web page	A	
	Explain the process of building and running Java application programs	A	
	Explain Java run time environment	U	
	Write a simple Java program and explain	A	
	Write a simple Java program to illustrate the use of mathematical functions	A	
	List the different sections of Java program structure	U	
I	List the advantages of OOPS.	U	
	Discuss OOPs areas of application.	A	
	Define the following. a. Data abstraction. b. Data encapsulation. c. Inheritance.	R	
	Define the following.	R	

	a. Polymorphism. b. Dynamic binding. c. Message Communication.		10
	List and explain Java features.	A	
	Discuss how Java differs from C & C++.	U	
	Explain the contributions of Java to the world wide web. With a figure illustrate how Java communicates with a web page.	U	
	Write a Java program to illustrate an application with two classes	A	
	Explain Java program structure with a diagram	U	
	Explain the features of Java.	A	
	List and explain Java statements.	U	
	Describe in detail the steps involved in implementing a standalone program	U	
	Explain Java virtual machine.	U	
	Discuss command line arguments in Java with an example program	U	
	With general syntax explain formatted input-output statements. Give example for each.	U	
II	Define constructors. List its special properties	U	05
	Define object. Explain object creation from a class	U	
	Define multilevel inheritance with an example	U	
	Define hierarchical inheritance with an example	U	
	Describe the syntax of single inheritance in Java	U	
	Compare and contrast overloading and overriding methods	U	
	Define subclass constructor and the use of keyword super	U	
	Define final classes and finalize methods	U	
	Define abstract methods and classes	U	
	Write the visibility control of Java classes.	U	
	Compare arrays and vectors	U	
	Explain any five string methods.	U	
	Explain any five string buffer methods	U	
	Explain the use of wrapper class in Java	U	
	Differentiate between interfaces and classes	U	
	Explain class definition with fields and methods declaration	U	10
	Discuss object creation in Java.	U	
	With an example explain accessing of class members	A	
	Write a program to illustrate constructor	A	
	Discuss the process of method overloading	U	
	Define static member. Write a program to illustrate static members.	A	
	Write a program to illustrate method overloading	A	
	Write a program to sort N elements of an array.	A	
	Write a program to illustrate vectors.	A	
	Write a program to illustrate wrapper classes	A	
III	Differentiate between interfaces and classes	U	05
	Write the general syntax of creating an interface and explain	U	
	Explain the various forms of interface implementation	U	
	Explain how to access interface variables with an example		
	Write the similarities between class and interfaces	U	

	Explain how to extend interfaces with an example	**A**	**10**
	With an example explain how to support multiple inheritance	**A**	
	Write a program to implement interfaces.	**A**	
IV	Define a package. List Java API packages	**U**	**05**
	How to create and implement a package	**U**	
	Explain the naming convention of a package with an example	**U**	
	Explain how to access packages with an example	**A**	
	Explain how to add a class to a package with an example	**A**	
	Discuss the various levels of access protection available for packages	**U**	
	Explain how to hide classes in a package	**U**	
	Explain static import and how is it useful	**U**	
	Explain Java API packages.	**U**	**10**
	Write a program to add a class to a package.	**A**	
	Write a Package program to demonstrate basic arithmetic operators	**A**	
	Write a program to use inbuilt packages to calculate square root of a number	**A**	
V	Define thread.	**R**	**05**
	Differentiate between multithreading and multitasking	**U**	
	Explain how to create thread by extending Thread class with an example	**U**	
	Explain how to stop and block a thread	**U**	
	Describe complete life cycle of a thread	**U**	
	Explain yield(),sleep() and stop() methods of a thread	**U**	
	Write a note on thread exceptions	**U**	
	How do we set priorities for threads?	**U**	
	How to create a runable thread?	**U**	
	Define synchronization? When do we use it	**U**	
	Differentiate between suspending and stopping a thread	**U**	
	Explain the different methods of creating threads	**U**	**10**
	Write a program to create a threads using a thread class	**A**	
	Explain thread creation by implementing runnable interface with an example	**A**	
	Write a note on inter- thread communication	**U**	
	Write a program to create threads by extending thread class	**A**	
	Write a program to create threads by implementing runnable interface	**A**	
	Write a program to set priorities for threads	**A**	
	Write a program to implement yield(),sleep() and stop() methods of a thread	**A**	
	Explain the life cycle of thread.	**U**	
VI	Define exception and explain its purpose	**U**	**05**
	Explain types of errors with suitable examples	**U**	
	List the compile time errors	**U**	
	List the run time errors	**U**	
	Illustrate compile time errors with an example	**A**	
	Illustrate run time errors with an example	**A**	
	Explain the syntax of exception handling	**U**	
	Write a program to illustrate nested try statements	**A**	

	Explain nested try statements with an example	**A**	
	Explain multiple catch blocks with an example	**A**	
	How many catch blocks can be used with one try block.,explain	**U**	
	Create a try block that is likely to generate three types of exception and then incorporate necessary catch block to catch and handle them appropriately	**A**	
	Explain the finally block. When and how it is used with a suitable example	**U**	
	Explain how exception handling mechanism can be used for debugging a program	**A**	
	Define an exception called "No MatchException"that is thrown when a string is not equal to " İndia".Write a program that uses this exception	**A**	
	Explain how to throw our own exceptions	**U**	
	Write a program to implement " Throwing our own exceptions"	**A**	
	Write a program to illustrate multiple catch blocks	**A**	**10**
	Write a program to use multiple catch block statement.	**A**	
	Write a program to illustrate nested try statement.	**A**	
	Write a program for throwing your own exception	**A**	