

Program Name	: Computer Engineering Program Group
Program Code	: CO/CM/IF/CW
Semester	: Fourth
Course Title	: Java Programming
Course Code	: 22412

1. RATIONALE

Java is platform independent, open-source object oriented programming language enriched with free and open source libraries. In current industrial scenario Java has the broad industry support and is prerequisite with many allied technologies like Advanced Java, Java Server Pages, and Android Application Development. Thus, current industrial trends necessitate acquiring Java knowledge for Computer Engineering and Information Technology graduates. This course develops necessary skills in students to apply object oriented programming techniques in Java so that students will be able to develop complete applications using core Java.

2. COMPETENCY

The aim of this course is to help the student to attain the following industry identified competency through various teaching learning experiences:

- **Develop Applications using Java.**

3. COURSE OUTCOMES (COs)

The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following *industry oriented* COs associated with the above mentioned competency:

- a. Develop programs using Object Oriented methodology in Java.
- b. Apply concept of inheritance for code reusability.
- c. Develop programs using multithreading.
- d. Implement Exception Handling.
- e. Develop programs using graphics and applet.
- f. Develop programs for handling I/O and file streams.

4. TEACHING AND EXAMINATION SCHEME

Teaching Scheme			Credit (L+T+P)	Examination Scheme												
L	T	P		Theory						Practical						
				Paper Hrs.	ESE		PA		Total		ESE		PA		Total	
3	-	4	7	3	70	28	30*	00	100	40	50#	20	50	20	100	40

(*): Under the theory PA; Out of 30 marks, 10 marks of theory PA are for micro-project assessment to facilitate integration of COs and the remaining 20 marks is the average of 2 tests to be taken during the semester for the assessment of the UOs required for the attainment of the COs.

Legends: L-Lecture; T – Tutorial/Teacher Guided Theory Practice; P -Practical; C – Credit, ESE -End Semester Examination; PA - Progressive Assessment.

5. COURSE MAP (with sample COs, PrOs, UOs, ADOs and topics)



This course map illustrates an overview of the flow and linkages of the topics at various levels of outcomes (details in subsequent sections) to be attained by the student by the end of the course, in all domains of learning in terms of the industry/employer identified competency depicted at the centre of this map.

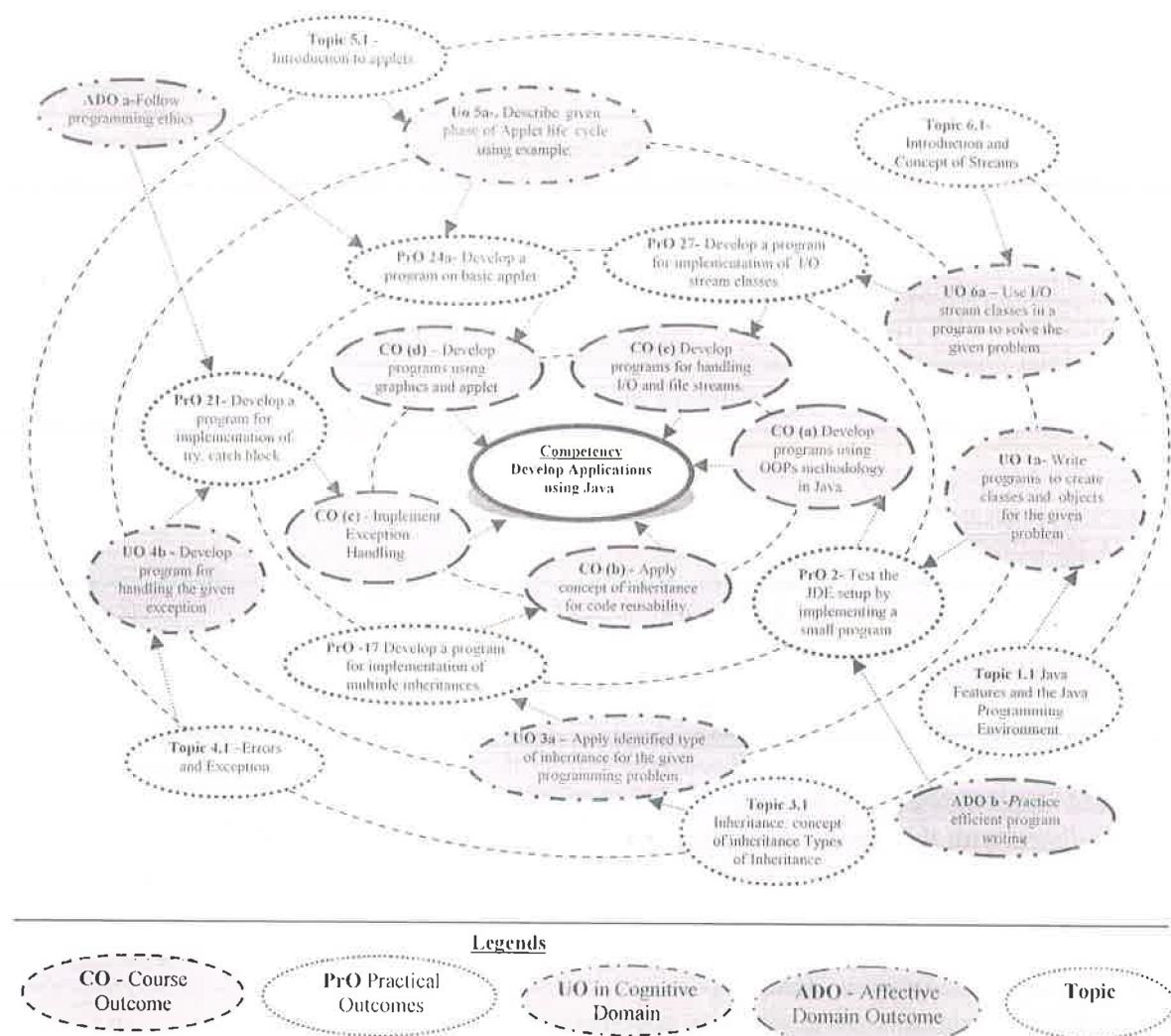


Figure 1 - Course Map

6. SUGGESTED PRACTICALS/ EXERCISES

The practicals in this section are PrOs (i.e. sub-components of the COs) to be developed and assessed in the student for the attainment of the competency.

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
1.	Setup a Java Programming development environment by using: a) Command prompt.(Classpath and path setup) b) Any IDE (Eclipse, Jcreator etc.).	I	02*
2.	Test the JDE setup by implementing a small program.	I	02
3.	Develop programs to demonstrate use of if statements and its different forms.	I	02*
4.	Develop programs to demonstrate use of a) Switch – Case statement b) Conditional if (? :)	I	02*
5.	Develop programs to demonstrate use of Looping Statement ‘for’	I	02*

S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
6.	Develop programs to demonstrate use of 'while', 'do-while'	I	02*
7.	Develop a program for implementation of implicit type casting in Java. Part-I	I	02
8.	Develop a program for implementation of implicit type casting in Java. Part-II	I	02
9.	Develop a program for implementation of explicit type conversion in Java.		02*
10.	a) Develop a program for implementation of Constructor. b) Develop a program for implementation of multiple constructors in a class.	II	02*
11.	Develop a program for implementation of different functions of String Class. Part-I	II	02
12.	Develop a program for implementation of different functions of String Class. Part-II	II	02
13.	Develop a program for implementation of Arrays in Java.	II	02*
14.	Develop a program for implementation of Vectors in Java.	II	02*
15.	Develop a program for implementation of Wrapper Class to convert primitive into object.	II	02*
16.	Develop a program for implementation of Wrapper Class to convert object into primitive.	II	02*
17.	Develop program which implements the concept of overriding.	II	02*
18.	Develop a program for implementation of Single and Multilevel inheritance.	III	02*
19.	Develop a program for implementation of multiple inheritances.	III	02*
20.	Develop a program to import different classes in package.	III	02*
21.	Develop a program for implementation of multithreading operation Part-I	IV	02*
22.	Develop a program for implementation of multithreading operation Part-II	IV	02
23.	Develop a program for implementation of try, catch block. Part-I	IV	02
24.	Develop a program for implementation of try, catch block. Part-II	IV	02
25.	Develop a program for implementation of try, catch and finally block.	IV	02*
26.	Develop programs for implementation of throw, throws clause. Part-I	IV	02*
27.	Develop programs for implementation of throw, throws clause. Part-II	IV	02*
28.	Develop minimum two basic Applets. Display output with applet viewer and browser. a) Develop a program on basic applet. b) Develop a program using control loops in applets.	V	02*
29.	Write a program to create animated shape using graphics and applets. You may use following shapes: a) Lines and Rectangles. b) Circles and Ellipses. c) Arcs d) Polygons with fillPolygon method.	V	02*
30.	Develop a program to draw following shapes, graphics and applets. a) Cone b) Cylinders	V	02*



S. No.	Practical Outcomes (PrOs)	Unit No.	Approx. Hrs. Required
	c) Cube		
	d) Square inside a circle e) Circle inside a square		02
31.	Develop a program for implementation of I/O stream classes.	VI	02*
32.	Develop a program for implementation of file stream classes.	VI	02*
	Total		64

Note

- i. A suggestive list of PrOs is given in the above table. More such PrOs can be added to attain the COs and competency. A judicious mix of minimum 24 or more practical need to be performed, out of which, the practicals marked as '*' are compulsory, so that the student reaches the 'Application Level' of Bloom's Taxonomy' as generally required by the industry.
- ii. The 'Process' and 'Product' related skills associated with each PrO are to be assessed according to a suggested sample given below:

Sr. No.	Performance Indicators	Weightage in %
1	Representation of classes and objects.	10
2	Coding.	40
4	Testing and Debugging of the Program.	30
5	Correctness of ProgramOutput.	10
6	Submission of report in time.	10
	Total	100

The above PrOs also comprise of the following social skills/attitudes which are Affective Domain Outcomes (ADOs) that are best developed through the laboratory/field based experiences:

- a. Follow safety practices.
- b. Practice good housekeeping.
- c. Demonstrate working as a leader/a team member.
- d. Follow ethical practices.

The ADOs are not specific to any one PrO, but are embedded in many PrOs. Hence, the acquisition of the ADOs takes place gradually in the student when s/he undertakes a series of practical experiences over a period of time. Moreover, the level of achievement of the ADOs according to Krathwohl's 'Affective Domain Taxonomy' should gradually increase as planned below:

- 'Valuing Level' in 1st year
- 'Organising Level' in 2nd year and
- 'Characterising Level' in 3rd year.

7. MAJOR EQUIPMENT/ INSTRUMENTS REQUIRED

The major equipment with broad specification mentioned here will usher in uniformity in conduct of experiments, as well as aid to procure equipment by authorities concerned.



S. No.	Equipment Name with Broad Specifications	Pro S.No.
1	Computer with JDK1.8 or above	All
2	Any IDE for Java Programming such as Eclipse, Jcreator or any other product.	

8. UNDERPINNING THEORY COMPONENTS

The following topics are to be taught and assessed in order to develop the sample UOs given below for achieving the COs to attain the identified competency. More UOs could be added.

Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
Unit – I Basic Syntactical constructs in Java	<p>1a. Write programs to create classes and objects for the given problem.</p> <p>1b. Explain the characteristics of the given Java token.</p> <p>1c. Explain the function of the given operator with example.</p> <p>1d. Construct the expressions using implicit and explicit type conversions to solve the given problem.</p> <p>1e. Develop the programs using relevant control structure to solve the given problem.</p>	<p>1.1 Java Features and the Java Programming Environment, Object Oriented, Compiled, Interpreted, Platform independent, Portable, Robust and Secure, Dynamic.</p> <p>1.2 Defining a class, creating object, accessing class members</p> <p>1.3 Java Tokens and Data types, Constants and Symbolic Constants, variables, dynamic initialization, data types, array and string, scope of variable, typecasting, and standard default values.</p> <p>1.4 Operators and Expressions, Arithmetic Operators, Relational Operators, Logical Operators, Increment and Decrement, Conditional Operators, Bit wise Operators, Instance of Operators, Dot Operators, Operator precedence and associativity, Evaluation of Expressions, Type conversions in expressions, Mathematical Functions - min(),max(), sqrt(), pow(), exp(), round(), abs().</p> <p>1.5 Decision making and looping: If statement, if else statement, nested if else statement, if else if ladder, the switch statement, nested switch statement, The ?:operator, The while statement, the Do while statement, the ‘for’ statement, break, continue and return statement, nested loops, labeled loops, for-each version of the for loop.</p>
Unit-II Derived Syntactical Constructs in Java	<p>2a. Use Constructors for the given programming problem.</p> <p>2b. Identify scope and lifetime of a variable in the given program code.</p> <p>2c. Describe the given visibility control with example.</p> <p>2d. Write the programs by implementing arrays to solve</p>	<p>2.1 Constructors and methods, types of constructors, nesting of methods, argument passing the ‘this’ keyword, command line arguments, varargs: variable-length arguments, garbage collection, finalize() method, the object class.</p> <p>2.2 Visibility Control Public, Private, Protected, default, friendly private</p>



Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
	<p>the given problem.</p> <p>2e. Develop programs using vectors and wrapper classes for the given problem.</p>	<p>Protected access.</p> <p>2.3 Arrays and Strings: Types of arrays, creating an array, strings, string classes and string buffer, vectors, wrapper classes, enumerated types.</p>
Unit– III Inheritance , Interface and Package	<p>3a. Apply the identified type of inheritance for the given programming problem.</p> <p>3b. Differentiate between overloading and overriding for the given example.</p> <p>3c. Develop program using the specified interface.</p> <p>3d. Create user defined package for the given problem.</p> <p>3e. Add class and interface to the given package.</p>	<p>3.1 Inheritance: concept of inheritance Types of Inheritance</p> <p>3.2 Single Inheritance, multilevel Inheritance, Hierarchical Inheritance, method and constructor overloading and overriding, dynamic method dispatch, final variables, final methods, use of super, abstract methods and classes, static members.</p> <p>3.3 Interfaces: Define Interface, implementing interface ,accessing interface, variables and methods, extending interfaces, interface references, nested interfaces</p> <p>3.4 Package: Define package, type of package naming and creating packages, accessing package, import statement, static import, adding class and interfaces to a package.</p>
Unit– IV Exception Handling and Multithreading	<p>4a. Distinguish the errors and exceptions (if any) in the given example.</p> <p>4b. Develop program for handling the given exception.</p> <p>4c. Crate threads to run the given multiple processes in the given program.</p> <p>4d. Explain the function of the specified phase in thread life cycle using the given example.</p>	<p>4.1 Errors and Exception :Types of errors, exceptions, try and catch statement, nested try statement, throws and Finally statement, build-in exceptions, chained exceptions, creating own exception(throw clause), subclasses.</p> <p>4.2 Multithreaded Programming Creating a Thread: By extending to thread class and by implementing runnable Interface, Life cycle of thread: Thread Methods:wait(), sleep(), notify(), resume(), suspend(), stop().Thread exceptions, thread priority and methods, synchronization, inter-thread communication, deadlock.</p>
Unit –V Java Applets and Graphics Programming	<p>5a. Describe the given phase of applet life cycle using a typical example.</p> <p>5b. Develop programs using applet implementation for the given problem.</p> <p>5c. Develop program for implementing the given geometric shape.</p> <p>5d. Develop program for implementing the given font</p>	<p>5.1 Introduction to applets: Applet, Applet life cycle (skeleton), Applet tag, Adding Applet to HTML file, passing parameter to applet, embedding <applet> tags in java code, adding controls to applets.</p> <p>5.2 Graphics Programming: Graphics classes, lines, rectangles, ellipse, circle, arcs, polygons, color and fonts, setColor(), getColor(), setForeground(), setBackground(), font class, variable defined by font class: name, pointSize,</p>



Unit	Unit Outcomes (UOs) (in cognitive domain)	Topics and Sub-topics
	settings.	size, style, font methods: getFamily(), getFont(), getFontname (), getSize(), getStyle(), getAllFonts() and get available font family name() of the graphics environment class.
Unit –VI Managing Input /Output/ Files in Java	6a. Use I/O stream classes in a program to solve the given problem. 6b. Write programs for reading and writing character streams to and from the given files. 6c. Write programs for reading and writing bytes to and from the given files. 6d. Write program to demonstrate use of primitive Data types with the specified stream.	6.1 Introduction and Concept of Streams. 6.2 Stream Classes. 6.3 Byte Stream Classes: Input Stream Classes, Output Stream Classes. 6.4 Character Stream Classes, Using streams. 6.5 Using File Class: I/O Exceptions, Creation of Files, Reading/Writing characters, Reading/Writing Bytes, Handling Primitive Data types.

Note: To attain the COs and competency, above listed UOs need to be undertaken to achieve the 'Application Level' of Bloom's 'Cognitive Domain Taxonomy'.

9. SUGGESTED SPECIFICATION TABLE FOR QUESTION PAPER DESIGN

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Basic Syntactical constructs in Java	06	02	04	04	10
II	Derived Syntactical Constructs in Java	10	02	06	10	18
III	Inheritance, Interface and Package	10	02	04	06	12
IV	Exception Handling and Multithreading	08	02	04	06	12
V	Java Applets and Graphics Programming	08	02	04	04	10
VI	Managing Input/Output/Files in Java	06	02	02	04	08
Total		48	12	24	34	70

Legends: R=Remember, U=Understand, A=Apply and above (Bloom's Revised taxonomy)

Note: This specification table provides general guidelines to assist students for their learning and to teachers to teach and assess students with respect to attainment of UOs. The actual distribution of marks at different taxonomy levels (of R, U and A) in the question paper may vary from above table.

10. SUGGESTED STUDENT ACTIVITIES

Other than the classroom and laboratory learning, following are the suggested student-related **co-curricular** activities which can be undertaken to accelerate the attainment of the various outcomes in this course: Students should conduct following activities in group and prepare reports of about 5 pages for each activity, also collect/record physical evidences for their (student's) portfolio which will be useful for their placement interviews:



- a. Prepare journals based on practical performed in laboratory.
- b. Follow coding standards.
- c. Develop variety of programs to improve the logical skills.
- d. Develop Application oriented real world programs.
- e. Prepare power point presentation or animation for understanding different Object Oriented Concepts.

11. SUGGESTED SPECIAL INSTRUCTIONAL STRATEGIES (if any)

These are sample strategies, which the teacher can use to accelerate the attainment of the various outcomes in this course:

- a. Massive open online courses (**MOOCs**) may be used to teach various topics/sub topics.
- b. '**L**' in item No. 4 does not mean only the traditional lecture method, but different types of teaching methods and media that are to be employed to develop the outcomes.
- c. About **15-20% of the topics/sub-topics** which is relatively simpler or descriptive in nature is to be given to the students for **self-directed learning** and assess the development of the COs through classroom presentations (see implementation guideline for details).
- d. With respect to item No.10, teachers need to ensure to create opportunities and provisions for **co-curricular activities**.
- e. Use different Audio Visual media for Concept understanding.
- f. Guide student(s) in undertaking micro-projects.
- g. Demonstrate students thoroughly before they start doing the practice.
- h. Observe continuously and monitor the performance of students in Lab.

12. SUGGESTED MICRO-PROJECTS

Only one micro-project is planned to be undertaken by a student that needs to be assigned to him/her in the beginning of the semester. In the first four semesters, the micro-project are group-based. However, in the fifth and sixth semesters, it should be preferably be **individually** undertaken to build up the skill and confidence in every student to become problem solver so that s/he contributes to the projects of the industry. In special situations where groups have to be formed for micro-projects, the number of students in the group should **not exceed three**.

The micro-project could be industry application based, internet-based, workshop-based, laboratory-based or field-based. Each micro-project should encompass two or more COs which are in fact, an integration of PrOs, UOs and ADOs. Each student will have to maintain dated work diary consisting of individual contribution in the project work and give a seminar presentation of it before submission. The total duration of the micro-project should not be less than **16 (sixteen) student engagement hours** during the course. The student ought to submit micro-project by the end of the semester to develop the industry oriented COs.

A suggestive list of micro-projects are given here. Similar micro-projects could be added by the concerned faculty:

- a. Mini Banking System for handling deposits and withdrawal.
- b. Medical Store stock Management System.
- c. Library book issue Management System.
- d. Bus Reservation System.
- e. Attendance Management System.
- f. Develop a small animation using applet, graphics and multithreading.

GUIDELINES FOR DEVELOPING MICRO PROJECTS:

- i. Declare four to five classes and may include Interfaces if required.
- ii. Must use Most of the Object Oriented Concepts.



- iii. Must implement concepts of Inheritance and Exception Handling.
- iv. Must Create Own Package.
- v. May use the constructor overloading and overriding.
- vi. May Use Multithreading if required.

13. SUGGESTED LEARNING RESOURCES

S. No.	Title of Book	Author	Publication
1.	Programming with JAVA	Balagurusamy E.	Mcgraw Hill Education (India) Private Limited, New Delhi, 5th Edition ISBN-13: 978-93-5134-320-2
2.	Java 8 Programming Black Book	DT Editorial Services	Dreamtech Press, New Delhi, ISBN:978-93-5119-758-4
3.	Java Complete Reference	Schildt Herbert	Mcgraw Hill Education, New Delhi ISBN:9789339212094
4.	Advanced Java Programming	Roy Uttam K	Oxford University Press, New Delhi ISBN :0-19-945550-3
5.	Jawa Programming	Dr. Rajendra Kawale	Devraj Publication

14. SUGGESTED SOFTWARE/LEARNING WEBSITES

- a. <https://docs.oracle.com/javase/8/docs/>
- b. http://www.nptelvideos.com/java/java_video_lectures_tutorials.php
- c. <http://nptel.ac.in/courses/106105084/25>
- d. <http://www.iitk.ac.in/esc101/08Jul/notes.html>

