

Lovely Professional University, Punjab

Course Code	Course Title	Lectures	Tutorials	Practicals	Credits	
CSE310	PROGRAMMING IN JAVA	3	0	2	4	
Course Weightage	ATT: 5 CA: 50 ETP: 45	Exam Category: X6: Mid Term Exam: Not Applicable – End Term Exam: Practical				
Course Focus	EMPLOYABILITY,SKILL DEVELOPMENT,ENTREPRENEURSHIP					

Course Outcomes :Through this course students should be able to

CO1 :: explain basic constructs of Java programming and apply them to solve the real-world problems

CO2 :: Illustrate the Object-oriented programming principles to write efficient and reusable codes.

CO3 :: demonstrate the concept of inheritance to reuse and extend the features of existing class with access control

CO4 :: contrast the uses of abstract classes, interfaces and Lambda expressions

CO5 :: use of exception handling and input/output techniques to improve the robustness and reliability of Java applications

CO6 :: integrate collections and generics to ensure clean, robust, and maintainable Java code

	TextBooks (T)		
Sr No	Title	Author	Publisher Name
T-1	PROGRAMMING WITH JAVA: A PRIMER, 4E	E. BALAGURUSAMY	MCGRAW HILL EDUCATION

	Reference Books (R)		
Sr No	Title	Author	Publisher Name
R-1	INTRODUCTION TO JAVA PROGRAMMING	Y. DANIEL LIANG	PEARSON
R-2	JAVA THE COMPLETE REFERENCE	HERBERT SCHILDT	MCGRAW HILL EDUCATION

Relevant Websites (RW)		
Sr No	(Web address) (only if relevant to the course)	Salient Features
RW-1	https://docs.oracle.com/javase/8/	Updated and authentic documentation of latest features is available

Audio Visual Aids (AV)		
Sr No	(AV aids) (only if relevant to the course)	Salient Features
AV-1	http://freevideolectures.com/Course/2513/Java-Programming	Video lectures on basics constructs of Java Programming

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

AV-2	https://www.youtube.com/watch?v=8EOhMRM-BfE&list=PLd3UqWTnYXOnujVv13wiZfrFKUEg9jBeA	Basics elaborated with focus on professional certifications
------	---	---

Software/Equipments/Databases		
Sr No	(S/E/D) (only if relevant to the course)	Salient Features
SW-1	IntelliJ IDEA iDE	IDE to develop simple to complex application development in convenient and faster way.
SW-2	JDK 1.8, Command Prompt and Text Editor	For writing, compiling and testing basic java constructs

LTP week distribution: (LTP Weeks)	
Weeks before MTE	7
Weeks After MTE	7
Spill Over (Lecture)	7

Detailed Plan For Lectures

Week Number	Lecture Number	Broad Topic(Sub Topic)	Chapters/Sections of Text/reference books	Other Readings, Relevant Websites, Audio Visual Aids, software and Virtual Labs	Lecture Description	Learning Outcomes	Pedagogical Tool Demonstration/ Case Study / Images / animation / ppt etc. Planned	Live Examples
Week 1	Lecture 1	Introduction to Java(History and Features of Java, Java program structure, Writing simple Java class and main() method, Command-line arguments, Understanding JDK, JRE and JVM)	T-1 R-2	RW-1 SW-1 SW-2 AV-2	Writing first java program, and using command line arguments.	Understanding the importance of Java programs	Lecture and discussion	Demonstration of Basic Java program along with JDK architecture
	Lecture 2	Introduction to Java(History and Features of Java, Java program structure, Writing simple Java class and main() method, Command-line arguments, Understanding JDK, JRE and JVM)	T-1 R-2	RW-1 SW-1 SW-2 AV-2	Writing first java program, and using command line arguments.	Understanding the importance of Java programs	Lecture and discussion	Demonstration of Basic Java program along with JDK architecture

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

Week 1	Lecture 3	Introduction to Java(History and Features of Java, Java program structure, Writing simple Java class and main() method, Command-line arguments, Understanding JDK, JRE and JVM)	T-1 R-2	RW-1 SW-1 SW-2 AV-2	Writing first java program, and using command line arguments.	Understanding the importance of Java programs	Lecture and discussion	Demonstration of Basic Java program along with JDK architecture
Week 2	Lecture 4	Data In the Cart(Using primitive data types, Type conversion, Keywords, Identifiers, Variables, Access modifiers, static keyword, Wrapper class)	T-1 R-2	SW-1 SW-2	Working with Primitive data types, keywords, variables, and access modifiers.	Understanding the usage of different data types and Narrowing/ Widening and importance of static in Java and Autoboxing and Unboxing.	Lecture cum demonstration and Discussion	Programs to demonstrate different data types, utoboxing and unboxing
	Lecture 5	Data In the Cart(Using primitive data types, Type conversion, Keywords, Identifiers, Variables, Access modifiers, static keyword, Wrapper class)	T-1 R-2	SW-1 SW-2	Working with Primitive data types, keywords, variables, and access modifiers.	Understanding the usage of different data types and Narrowing/ Widening and importance of static in Java and Autoboxing and Unboxing.	Lecture cum demonstration and Discussion	Programs to demonstrate different data types, utoboxing and unboxing
	Lecture 6	Data In the Cart(Using primitive data types, Type conversion, Keywords, Identifiers, Variables, Access modifiers, static keyword, Wrapper class)	T-1 R-2	SW-1 SW-2	Working with Primitive data types, keywords, variables, and access modifiers.	Understanding the usage of different data types and Narrowing/ Widening and importance of static in Java and Autoboxing and Unboxing.	Lecture cum demonstration and Discussion	Programs to demonstrate different data types, utoboxing and unboxing
Week 3	Lecture 7	Operators(Working with Bit-wise, arithmetic, logical, and relational operators, Unary, assignment and Ternary operator, Operator precedence)	T-1	SW-1 SW-2	Using various operators and operator precedence	Understanding the working of Bit-wise operators, short-circuit operators and operator precedence	Lecture cum demonstration and Discussion	Program to demonstrate the working of different operators
	Lecture 8	Conditional Statements (Using if/else constructs and switch-case statements)	T-1 R-1	SW-1 SW-2 AV-1	Using if-else if-else ladder and switch statement	Understanding different ways to work with conditional statements	Lecture cum demonstration and Brainstorming	Program to display the grades of the student based on the marks
	Lecture 9	Loops(Working with for loop, while loop, do-while loop and for-each loop)	T-1 R-1	SW-1 SW-2 AV-1	Using different types of loops in Java	Working with for, while, do-while and for-each loop	Lecture cum demonstration and Discussion	Program to implement various loops in Java

Week 4	Lecture 10	Arrays and Enums (Fundamentals about Arrays, Multi-dimensional arrays, Array Access and Iterations, Using varargs, Enumerations)	R-1 R-2	SW-2	Creation and usage of one dimensional and two-dimensional arrays	Using one-dimensional, two-dimensional arrays and varargs	Lecture cum demonstration and Brainstorming	Program to find the grades of student with different number of academic tasks
	Lecture 11	OOP Concepts(Basics of class and objects, Writing constructors and methods, Overloading methods and constructors, this keyword, initializer blocks)	T-1		Creating classes, initializing objects. Implementing method and constructor overloading. Using this keyword. Implementing Initializer blocks.	Understanding implementation of OOP concepts.	Lecture cum demonstration and Discussion	Programs implementing different oop concepts.
	Lecture 12	OOP Concepts(Basics of class and objects, Writing constructors and methods, Overloading methods and constructors, this keyword, initializer blocks)	T-1		Creating classes, initializing objects. Implementing method and constructor overloading. Using this keyword. Implementing Initializer blocks.	Understanding implementation of OOP concepts.	Lecture cum demonstration and Discussion	Programs implementing different oop concepts.
Week 5	Lecture 13	OOP Concepts(Basics of class and objects, Writing constructors and methods, Overloading methods and constructors, this keyword, initializer blocks)	T-1		Creating classes, initializing objects. Implementing method and constructor overloading. Using this keyword. Implementing Initializer blocks.	Understanding implementation of OOP concepts.	Lecture cum demonstration and Discussion	Programs implementing different oop concepts.
	Lecture 14	String Class(Constructors and methods of String and String Builder class)	T-1 R-1	RW-1 SW-2	Implementing the constructors and methods of String and StringBuilder class	Understanding different ways of creating Strings and using the functionalities. Using mutable String objects.	Lecture cum demonstration and Discussion	Program to implement the methods of String and StringBuilder
	Lecture 15	String Class(Constructors and methods of String and String Builder class)	T-1 R-1	RW-1 SW-2	Implementing the constructors and methods of String and StringBuilder class	Understanding different ways of creating Strings and using the functionalities. Using mutable String objects.	Lecture cum demonstration and Discussion	Program to implement the methods of String and StringBuilder
Week 6	Lecture 16				Test - Code based 1			

Week 6	Lecture 17	Inheritance and Polymorphism(Inheritance, Method overriding, super keyword, Object class and overriding toString() and equals() method, Using super and final keywords, instanceof operator)		RW-1 SW-1	Inheritance in java	Understanding the importance of IS-A relationship, Polymorphism and keywords in inheritance	Lecture cum demonstration and Discussion	Program to create subclasses and implement different keywords.
	Lecture 18	Inheritance and Polymorphism(Inheritance, Method overriding, super keyword, Object class and overriding toString() and equals() method, Using super and final keywords, instanceof operator)		RW-1 SW-1	Inheritance in java	Understanding the importance of IS-A relationship, Polymorphism and keywords in inheritance	Lecture cum demonstration and Discussion	Program to create subclasses and implement different keywords.
Week 7	Lecture 19	Abstract Class and Interface (Abstract method and abstract class, Interfaces, static and default methods, Using Swing Components to demonstrate inheritance)	T-1 R-1	RW-1 SW-1	Implementing abstraction using abstract class and abstract methods. Implementing interfaces	Understanding the importance of abstract method, abstract class and interfaces	Lecture cum demonstration and Discussion	Program to implement abstract class and its inheritance
SPILL OVER								
Week 7	Lecture 20				Spill Over			
	Lecture 21				Spill Over			
MID-TERM								
Week 8	Lecture 22	Functional Interface and Lambda Expressions(Using Lambda expressions, Implementing Threads using Lambda expressions, Implementing Listener using Lambda expressions)	R-1	SW-1 SW-2	Introduction of Functional Interface and Lambda Expressions.	Understanding the structure and usage of functional interfaces and Lambda Expressions	Lecture cum demonstration and Brainstorming	Program to demonstrate Lambda expressions
	Lecture 23	Functional Interface and Lambda Expressions(Using Lambda expressions, Implementing Threads using Lambda expressions, Implementing Listener using Lambda expressions)	R-1	SW-1 SW-2	Introduction of Functional Interface and Lambda Expressions.	Understanding the structure and usage of functional interfaces and Lambda Expressions	Lecture cum demonstration and Brainstorming	Program to demonstrate Lambda expressions

Week 8	Lecture 24	Functional Interface and Lambda Expressions(Using Lambda expressions, Implementing Threads using Lambda expressions, Implementing Listener using Lambda expressions)	R-1	SW-1 SW-2	Introduction of Functional Interface and Lambda Expressions.	Understanding the structure and usage of functional interfaces and Lambda Expressions	Lecture cum demonstration and Brainstorming	Program to demonstrate Lambda expressions
Week 9	Lecture 25	Nested Class(Understanding the importance of static and non-static nested classes, Local and Anonymous class)	T-1 R-1 R-2	RW-1 SW-1 SW-2	Implementing static, non-static nested classes, and local and anonymous classes.	Understanding the need of nested classes	Lecture cum demonstration and Brainstorming	Program to demonstrate the need of Nested Classes
	Lecture 26	Nested Class(Understanding the importance of static and non-static nested classes, Local and Anonymous class)	T-1 R-1 R-2	RW-1 SW-1 SW-2	Implementing static, non-static nested classes, and local and anonymous classes.	Understanding the need of nested classes	Lecture cum demonstration and Brainstorming	Program to demonstrate the need of Nested Classes
	Lecture 27	Utility Classes(Working with Dates)	T-1 R-2	RW-1 SW-1 SW-2	Using Dates in Java	Using Date in Java applications	Lecture cum demonstration and Discussion	Program to display date and time components in Java applications
Week 10	Lecture 28				Test - Code based 2			
	Lecture 29	Exceptions and Assertions (Exception overview, Exception class hierarchy and exception types, Propagation of exceptions, Using try, catch and finally for exception handling, Usage of throw and throws, handling multiple exceptions using multi-catch, Autoclose resources with try-with resources statement, Creating custom exceptions, Testing invariants by using assertions)	T-1	RW-1 SW-1 SW-2	Overview of Exception and its class hierarchy. Handling exceptions using try, catch and finally. Using multicatch, throw and throws.	Understanding different types of Exceptions and propagation of exception	Lecture cum demonstration and Discussion	Program to demonstrate the exception and its effects

Week 10	Lecture 30	Exceptions and Assertions (Exception overview, Exception class hierarchy and exception types, Propagation of exceptions, Using try, catch and finally for exception handling, Usage of throw and throws, handling multiple exceptions using multi-catch, Autoclose resources with try-with resources statement, Creating custom exceptions, Testing invariants by using assertions)	T-1	RW-1 SW-1 SW-2	Overview of Exception and its class hierarchy. Handling exceptions using try, catch and finally. Using multicatch, throw and throws.	Understanding different types of Exceptions and propagation of exception	Lecture cum demonstration and Discussion	Program to demonstrate the exception and its effects
Week 11	Lecture 31	Exceptions and Assertions (Exception overview, Exception class hierarchy and exception types, Propagation of exceptions, Using try, catch and finally for exception handling, Usage of throw and throws, handling multiple exceptions using multi-catch, Autoclose resources with try-with resources statement, Creating custom exceptions, Testing invariants by using assertions)	T-1	RW-1 SW-1 SW-2	Overview of Exception and its class hierarchy. Handling exceptions using try, catch and finally. Using multicatch, throw and throws.	Understanding different types of Exceptions and propagation of exception	Lecture cum demonstration and Discussion	Program to demonstrate the exception and its effects
	Lecture 32	Exceptions and Assertions (Exception overview, Exception class hierarchy and exception types, Propagation of exceptions, Using try, catch and finally for exception handling, Usage of throw and throws, handling multiple exceptions using multi-catch, Autoclose resources with try-with resources statement, Creating custom exceptions, Testing invariants by using assertions)	T-1	RW-1 SW-1 SW-2	Overview of Exception and its class hierarchy. Handling exceptions using try, catch and finally. Using multicatch, throw and throws.	Understanding different types of Exceptions and propagation of exception	Lecture cum demonstration and Discussion	Program to demonstrate the exception and its effects

Week 11	Lecture 33	Exceptions and Assertions (Exception overview, Exception class hierarchy and exception types, Propagation of exceptions, Using try, catch and finally for exception handling, Usage of throw and throws, handling multiple exceptions using multi-catch, Autoclose resources with try-with resources statement, Creating custom exceptions, Testing invariants by using assertions)	T-1	RW-1 SW-1 SW-2	Overview of Exception and its class hierarchy. Handling exceptions using try, catch and finally. Using multicatch, throw and throws.	Understanding different types of Exceptions and propagation of exception	Lecture cum demonstration and Discussion	Program to demonstrate the exception and its effects
Week 12	Lecture 34	I/O Fundamentals (Describing the basics of input and output in Java, Read and write data from the console, Using streams to read and write files, Writing and read objects using serialization)	T-1 R-1	RW-1 SW-1 SW-2	Use of Scanner, PrintWriter, FileInputStream, FileOutputStream, ObjectInputStream and ObjectOutputStream	Understanding how IO works in Java	Lecture cum demonstration and Discussion	Programs to demonstrate IO in Java
	Lecture 35	I/O Fundamentals (Describing the basics of input and output in Java, Read and write data from the console, Using streams to read and write files, Writing and read objects using serialization)	T-1 R-1	RW-1 SW-1 SW-2	Use of Scanner, PrintWriter, FileInputStream, FileOutputStream, ObjectInputStream and ObjectOutputStream	Understanding how IO works in Java	Lecture cum demonstration and Discussion	Programs to demonstrate IO in Java
	Lecture 36	I/O Fundamentals (Describing the basics of input and output in Java, Read and write data from the console, Using streams to read and write files, Writing and read objects using serialization)	T-1 R-1	RW-1 SW-1 SW-2	Use of Scanner, PrintWriter, FileInputStream, FileOutputStream, ObjectInputStream and ObjectOutputStream	Understanding how IO works in Java	Lecture cum demonstration and Discussion	Programs to demonstrate IO in Java
Week 13	Lecture 37				Test - Code based 3			

Week 13	Lecture 38	Collections and Generics (Creating a custom generic class, Using the type inference diamond to create an object, Using bounded types and Wild Cards, Creating a collection by using generics, Implementing an ArrayList, Implementing TreeSet using Comparable and Comparator interfaces, Implementing a HashMap, Implementing a Deque)	T-1 R-1	RW-1 SW-1 SW-2	Differentiating between Comparable and Comparator, use of TreeSet, HashMap, LinkedList	Implementing generic classes and collections.	Lecture cum demonstration and Discussion	Program to demonstrate the need of generic classes and their implementation
	Lecture 39	Collections and Generics (Creating a custom generic class, Using the type inference diamond to create an object, Using bounded types and Wild Cards, Creating a collection by using generics, Implementing an ArrayList, Implementing TreeSet using Comparable and Comparator interfaces, Implementing a HashMap, Implementing a Deque)	T-1 R-1	RW-1 SW-1 SW-2	Differentiating between Comparable and Comparator, use of TreeSet, HashMap, LinkedList	Implementing generic classes and collections.	Lecture cum demonstration and Discussion	Program to demonstrate the need of generic classes and their implementation
Week 14	Lecture 40	Collections and Generics (Creating a custom generic class, Using the type inference diamond to create an object, Using bounded types and Wild Cards, Creating a collection by using generics, Implementing an ArrayList, Implementing TreeSet using Comparable and Comparator interfaces, Implementing a HashMap, Implementing a Deque)	T-1 R-1	RW-1 SW-1 SW-2	Differentiating between Comparable and Comparator, use of TreeSet, HashMap, LinkedList	Implementing generic classes and collections.	Lecture cum demonstration and Discussion	Program to demonstrate the need of generic classes and their implementation
		SPILL OVER						
Week 14	Lecture 41				Spill Over			
	Lecture 42				Spill Over			
Week 15	Lecture 43				Spill Over			
	Lecture 44				Spill Over			

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

Week 15	Lecture 45				Spill Over			
---------	------------	--	--	--	------------	--	--	--

Scheme for CA:

CA Category of this Course Code is:C010203 (Total 4 tasks, 1 compulsory and out of remaining 2 best out of 3 to be considered)

Component	Iscompulsory	Weightage (%)	Mapped CO(s)
Test - Code based 1	NO	30	CO5, CO6
Test - Code based 2	NO	30	CO1, CO2
Test - Code based 3	NO	30	CO3, CO4
Programming Practice	Yes	40	CO1, CO2, CO3, CO4, CO5, CO6

Details of Academic Task(s)

Academic Task	Objective	Detail of Academic Task	Nature of Academic Task (group/individuals)	Academic Task Mode	Marks	Allotment / submission Week
Programming Practice	To evaluate the overall learnings of students.	<p>The students will be given approx. 150 coding problems and 150 MCQs on the selected third party platform. The questions will be framed with equal distribution from the complete syllabus. In order to qualify for programming practice marks, the student should solve at least 50% of the coding problems and 50% of MCQ questions (eligibility condition).</p> <p>The maximum marks out of 20 marks for which the student would be eligible for Programming Practice would be based on the Percentage of questions solved by the student.</p> <p>Example – If a student solves 105 questions out of 150 questions (i.e. 70% questions solved) then the student would be eligible for 70% of 20 marks which is 14 marks (round up would be used in case of decimal values).</p> <p>The final marks for Programming Practice would be calculated by prorating the eligible marks for which the student is eligible (as explained in the above point) with the percentage of marks student has scored in the proctored Coding Contests conducted as CA's (The final marks would be round up for the students).</p> <p>Example – If a student solves 105 questions out of 150 questions (i.e. 70% questions solved) then the student would be eligible for 70% of 20 marks which is 14 marks. And the student has scored 24 out of 30 in the CA's i.e. 80% marks in CA, his Programming practice final marks would be 80% of 14 marks that he was eligible for which is 11.2 rounded up to 12 marks out of 20 for Programming Practice.</p>	Individual	Online	20	1 / 14
Test - Code based 2	To evaluate basic programming skills of students	Mix of MCQs (10 Marks) and Coding Problems (20 Marks)	Individual	Online	30	4 / 5
Test - Code based 3	To evaluate programming and logic building skills of students	Mix of MCQs (10 Marks) and Coding Problems (20 Marks)	Individual	Online	30	9 / 10
Test - Code based 1	To evaluate programming and logic building skills of students	Mix of MCQs (10 Marks) and Coding Problems (20 Marks)	Individual	Online	30	12 / 13

Detailed Plan For Practicals

An instruction plan is only a tentative plan. The teacher may make some changes in his/her teaching plan. The students are advised to use syllabus for preparation of all examinations. The students are expected to keep themselves updated on the contemporary issues related to the course. Upto 20% of the questions in any examination/Academic tasks can be asked from such issues even if not explicitly mentioned in the instruction plan.

Practical No	Broad topic	Subtopic	Other Readings	Learning Outcomes
Practical 1	Creating a Java Main Class	Program to implement a java class.		Understanding the main() method and how command-line arguments work in Java.
Practical 2	Managing Multiple Items	Program to demonstrate the use of list of items.		Understanding different ways to work with conditional and looping statements
Practical 3	Manipulating and Formatting the Data in Your Program	Program to demonstrate the uses of String and StringBuilder		Understanding different ways of creating Strings and using the functionalities
Practical 4	Describing Objects and Classes	Program to demonstrate the instantiation of class and accessing the attributes using object of class.		Understanding the Object initialization and associating functionalities
Practical 5	Using Inheritance	Program to demonstrate the inheritance and its importance using Swing Components.		Understanding the importance of IS-A relationship and Polymorphism
Practical 6	Overriding Methods, Polymorphism, and Static Classes	Program to implement polymorphism and using proper access control.		Understanding the importance of IS-A relationship and Polymorphism
Practical 7	Abstract and Nested Classes	Program to demonstrate the use of abstract class and nested class.		Identifying the need of interfaces and static, default and abstract methods
Practical 8	Abstract and Nested Classes	Program to demonstrate the use of abstract class and nested class.		Identifying the need of interfaces and static, default and abstract methods
Practical 9	Multithreading	Program to implement multithreading using Lambda Expressions.		Using java multithreading concept to make subtasks run in parallel.
Practical 10	Multithreading	Program to implement multithreading using Lambda Expressions.		Using java multithreading concept to make subtasks run in parallel.
Practical 11	Exception Handling	Program to demonstrate the use of all the keywords used for exception handling and need of assertion		Understanding the exception handling using the try, catch and finally blocks
Practical 12	Exception Handling	Program to demonstrate the use of all the keywords used for exception handling and need of assertion		Understanding the exception handling using the try, catch and finally blocks
Practical 13	Java IO	Program to implement read and write operation using console and File.		Understanding how IO works in Java
Practical 14	Java IO	Program to implement read and write operation using console and File.		Understanding how IO works in Java
	SPILL OVER			
Practical 15	Spill Over			