CHENNAI
INSTITUTE OF TECHNOLOGY
(Autonomous)

(An Autonomous Institution, Affiliated to Anna University, Chennai)

DEPARTMENT OF Artificial Intelligence and Data Science

III- Semester

AD2305 – ADVANCED JAVA PROGRAMMING

(B.E/B.Tech – CSE/IT/AIDS/AIML/ CS)

	UNIT-I (FUNDAMENTALS OF JAVA) **PART A**
1.	Express what is meant by Object Oriented Programming
2.	Compare class and object.
3.	List the core OOP's concepts.
4.	Tabulate the differences between C++ and Java.
5.	List the Buzzwords of Java.
6.	Discuss, what is meant by abstraction.
7.	Describe about Encapsulation, Inheritance and Polymorphism.
8.	How is encapsulation and abstraction implemented in java?
9.	Create a simple Java Program to find whether the given number is Prime or not.
10.	Evaluate the characteristics of objects.
11	"Java is platform independent". Justify.
12	How to create multidimensional arrays in java?
13	Illustrate the working of Java Virtual Machine (JVM).
14	Write the command to compile and execute a java program in command prompt.
15	Write a Java program to check whether the given number is odd or even.
16	Describe the control flow statements in Java.
17	Write a Java program to find the largest three numbers in an array.
18	Write a Java program to sort the given numbers in ascending order.
19	Write a java program to generate fibonacci series.
20	Explain Ternary operator with an example.

UNIT-I **PART – B**		
1.	Explain OOPS and its features.	(13)
2.	Describe variables and operators in Java.	(13)
3.	Define and explain control flow statements in Java with suitable examples.	(13)
4.	a) Develop a simple Java program to sort the given numbers in increasing order. b) Write a Java program to reverse the given number.	(13)
5.	a) Classify the characteristics of Java. b) Illustrate the working principles of JVM.	(13)
6.	Define Arrays. What is array sorting and explain with an example.	(13)
7.	Explain about Parentheses and Operator hierarchy in Java.	(13)
8.	Discuss in detail about datatypes, variables, constants with example.	(13)
9.	Explain with example on Structure of a program in Java.	(13)
10.	Explain the concept of 2D array and looping constructs with an example program.	(13)
11.	a) Elaborate Java buzzwords in detail. b) Discuss about the various operators in Java with examples	(13)
12.	a) Explain in detail on Java Programming Environment set up. b) Develop a java program to compute factorial of a number.	(13)
13.	a) Demonstrate Bitwise operators using a Java program. b) Demonstrate the operator precedence using a Java program.	(13)
14.	a) Demonstrate Nested if-else using an example program. b) Demonstrate implicit and explicit type casting with an example program.	(13)
15.	a) List and explain the Unary, Binary and Ternary operators with examples. b) Develop a program to compute the inverse of a given matrix.	(13)
PART – C		
1.	Evaluate a Java program to find a smallest number in the given array by creating one dimensional array and two-dimensional array using new operator.	(15)
2.	Describe in detail about features of Object-Oriented Programming.	(15)
3.	Write a java program to print the information of students along with the grades of different subjects using control flow statements.	(15)
4.	a) Write a java program to compute matrix multiplication using a 3*3 array. b) Write a java program to pass arrays as parameters in a method and print the array elements	(15)
5.	Write a Java program for the calculator based on the following requirements: 1. Implement a control flow loop that displays a menu to the user with the following options: • Addition (+)	(15)

	• Subtraction (-) • Multiplication (*) • Division (/) • Quit the calculator.	

	UNIT-II (IMPLEMENTATION OF OOP CONCEPTS) **PART A**
1.	Define class and object in Java.
2.	Evaluate the characteristics of objects.
3.	List the various access specifiers supported by java.
4.	Illustrate constructors in Java.
5.	Assess with example what is meant by parameterized constructor.
6.	Define static variable and static method.
7.	Explain what is meant by Java package.
8.	Infer how to import a single package.
9.	Give any four Java doc comments.
10.	Explain the importance of inheritance.
11	State the characteristics of constructor.
12	Name some of the types of inner classes.
13	What is default constructor? Illustrate with an example.
14	Identify what are the uses of using super keyword.
15	Illustrate how protected members in a subclass can be accessed in Java.
16	Compare between abstract class and interface.
17	Elaborate the use of final keyword.
18	Give an example for dynamic method dispatch.
19	State any two differences between interface and abstract classes.
20	What is interface? How does it support multiple inheritance in java
UNIT-II **PART – B**	

1.	a) Define class. How do you define a class in Java? b) Give some examples for inheritance and its types.	(13)
2.	What is meant by constructor? Discuss the types of constructors with example.	(13)
3.	a) Summarize about access specifier in Java. b) Describe the static variable and method and explain its types with example.	(13)
4.	Illustrate what is meant by package? How it is created and implemented in Java?	(13)
5.	Interpret with an example, what is method overloading and method overriding?	(13)
6.	Demonstrate any two types of inheritance with suitable examples.	(13)
7.	Discuss in detail in brief about object, class and its methods in Java with suitable Examples.	(13)
8.	a) Describe what is meant by interface. b) How is interface declared and implemented in Java. Give example.	(13)
9.	Demonstrate the different uses of super keywords with suitable examples.	(13)
10.	What is dynamic method dispatch? Give suitable program in java.	(13)
11.	a) With sample program explain the creation of packages. Accessing a package and hiding classes with packages. b) Describe the usage of static members and nesting members with suitable example programs in java.	(13)
12.	a) What is the importance of constructor? b) Write a java program for the super class Quadrilateral, with child classes Parallelogram, Square and Rectangle. Calculate area of square, rectangle and parallelogram.	(13)
13.	a) Design a class that represents a bank account and construct the methods to i) Assign initial values ii) Deposit an amount iii) Withdraw amount after checking balance iv) Display the name and balance. b) Do you need to use static keyword for the above bank account program? Explain.	(13)
14.	a) What is an Interface? Give the general form of an Interface and also discuss the implementation details of Interfaces. b) Write a program which specify that there are two classes Rectangle and Circle which implements the interface Shape and find the area of rectangle and circle.	(13)
15.	a) With an example program explain the concept of classes and nested classes in java. b) Why constructors do not have any return type? Explain it with proper example.	(13)
	PART – C	
1.	Explain inner classes and its types with suitable examples.	(15)
2.	Develop a a Java application with Employee class with Emp_name, Emp_id, Address, Mail_id, Mobile_no as members. Inherit the classes, Programmer, Assistant Professor, Associate Professor and Professor from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 97% of BP as DA,	(15)

	10 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.	
3.	Illustrate how to control class level and member level access in java.	(15)
4.	a) Write a java program to illustrate the use of final keyword. b) Explain with an example program how super keyword can be used in different ways.	(15)
5.	Develop a Java Program to create an abstract class name Shape that contains two integers and an empty method name print Area (). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contains only the method print Area () that prints the area of the given shape.	(15)

	UNIT-III (EXCEPTION HANDLING AND MULTITHREADING) **PART A**
1.	Interpret what is an Exception. What is its use?
2.	State the differences between throw and throws keyword.
3.	What is re-throwing an exception?
4.	Define uncaught exception.
5.	Summarize the task performed by exception handling.
6.	Differentiate exception and error.
7.	Classify the exception types with example.
8.	Draw the exception hierarchy.
9.	What are the two methods available in stack trace elements?
10.	Formulate the advantages of using exception handling.
11	Give the properties of thread.
12	Show the different states in thread.
13	Why synchronization is required in thread?
14	What is the need for thread?
15	List out the methods which implement inter thread communication.
16	Give the idea to achieve thread synchronization in Java.
17	Define multithreading.
18	Give the life cycle of thread.

19	What are the ways in which a thread can be created in java?	
20	Write about the advantages and disadvantages of multithreading in JAVA.	
	UNIT-III **PART – B**	
1.	Discuss in detail about exception handling constructs and write a program to illustrate Divide by zero exception.	(13)
2.	Describe the following concepts with example. a) Try-catch-throw paradigm b) Any two built-in exceptions	(13)
3.	Discuss about the syntax of catch and throw an exception with an example.	(13)
4.	Tabulate any five classes to support Exception handling in Java with an example for each.	(13)
5.	Describe in detail about multithread programming with example.	(13)
6.	Summarize the two types of thread implementation supported by Java. Give example.	(13)
7.	a) Define multithreading and give a small example in java. b) Describe the properties of thread in detail.	(13)
8.	a) Classify the errors and exception in Java. b) Illustrate about multiple catching exceptions with example.	(13)
9.	Summarize the following. a) Thread properties b) Daemon thread.	(13)
10.	What is meant by Thread Synchronization in JAVA programming? Why it is important? With an sample program, explain the JAVA’s language-level support for synchronizing threads.	(13)
11.	What is the importance of Exception Handling in Java? Define and distinguish between checked and unchecked exceptions and Illustrate with an example.	(13)
12.	a) Is it possible to Rethrow exceptions? Justify your answer. b) Illustrate the Hierarchy of standard Exception class with a neat sketch.	(13)
13.	a) Illustrate the Thread states transitions with a neat sketch. b) Demonstrate Thread prioritization with a Java Program.	(13)
14.	Discuss the concept of throw, throws and finally keyword.	(13)
15.	a) What do you mean by multithreading? Develop a simple application program to illustrate the use of multithreading. b) “Intercommunication between thread is relatively economical than processes” justify this statement.	(13)
	PART – C	
1.	Consider a scenario where one thread (producer) is producing integer data starting from 0 and another thread (consumer) consuming it. In addition, assume that the producer has to wait until the consumer has finished consuming it before it generates more data. Using inter thread communication concept implement the above producer and consumer.	(15)
2.	Explain how to create threads. Write a java program that prints numbers from 1 to 10 line by line after every 5 seconds.	(15)

3.	Define a custom exception class, e.g., InvalidAgeException, and use it to validate the age of a user. Write a program that throws this custom exception when the age provided is negative.	(15)
4.	Write programs to illustrate ArithmeticException, ArrayIndexOutOfBoundsException and NumberFormatException.	(15)
5.	Explain inter thread communication and suspending, resuming and stopping threads.	(15)

	UNIT-IV(I/O, GENERICS, STRING HANDLING) **PART A**
1.	What are two types of I/O streams?
2.	List the any five byte stream class.
3.	What are the classes involved in character stream for input and output operations.
4.	Define generic class.
5.	Write the syntax for simple generic class.
6.	Summarize the advantages of generic programming.
7.	Write the syntax for generic method.
8.	Illustrate any four character stream class.
9.	Point out the syntax of buffered reader to connect to the keyboard.
10.	What are streams? What are their advantages?
11	Write a Java program to demonstrate the use of readline method.
12	Write the code in Java using a printwriter class to handle console output.
13	State the differences between String and StringBuffer.
14	Summarize any two string handling methods in Java.
15	Explain the difference between the System.out and System.err streams.
16	Compare between equals() and ==.
17	Evaluate how do you handle type erasure in Java generics?
18	List out any four methods from StringBuilder class.
19	Define serialization in Java? How is it implemented?

20	Evaluate the need for generic programming.	
	UNIT-IV **PART – B**	
1.	Describe briefly about the features a) Byte stream I/O b) Character streams input/output	(13)
2.	Explain the following with example a) Reading console input b) Writing console output.	(13)
3.	a) Identify a Java program to read characters from the console. b) Identify a Java program to read strings from the console.	(13)
4.	Illustrate in brief about a) Reading from a file. b) Writing in a file.	(13)
5.	a) Formulate the motivations of generic programming. b) Develop a program to show generic class and methods with example.	(13)
6.	What are generics and Generic programming? Explain the necessity of Generic Programming with an example.	(13)
7.	Describe the following a) Generic class and Generic method with syntax and example b) Restrictions and Limitations of Generic Programming	(13)
8.	a) Describe in detail about bounded types with suitable example. b) List the inheritance rules for generic types with example.	(13)
9.	Discuss in detail about common methods used for manipulating strings in Java.	(13)
10.	Introduce the StringBuilder class in Java. How does it differ from StringBuffer, and in what situations would you prefer to use StringBuilder over StringBuffer?	(13)
11.	In Java, what is the role of generics in achieving type safety? Provide a Java code example that uses generics to illustrate type safety.	(13)
12.	Explain the concept of type erasure in Java generics. How does type erasure impact the runtime behavior of generic code? Provide a code example to illustrate this.	(13)
13.	Discuss the differences between String and StringBuffer in Java. Provide scenarios where using StringBuffer is more efficient than using regular String objects.	(13)
14.	Write a Java program that uses StringBuilder to efficiently concatenate a large number of strings. Explain why StringBuilder is a better choice for such operations compared to regular string concatenation.	(13)
15.	Explain various methods of StringBuffer with an example.	(13)
	PART – C	
1.	Develop a simple generic class example with two type parameters. so that we can define two types of parameters called U & V, seperated by ",".	(15)

2.	Assess an example program in Java on how to implement bounded types with generics.	(15)
3.	Develop the Java program to concatenate the content of two files and produce the output in the third file.	(15)
4.	Deduce a Java program that reads a file name from the user, displays information about whether the file exists, whether the file is readable, or writable, the type of file and the length of the file in bytes.	(15)
5.	Write a Java program to find the maximum value from the given type of elements using a generic method.	(15)

	UNIT-V(COLLECTIONS FRAMEWORK & DATABASE CONNECTIVITY) **PART A**
1.	Define Java Collections Framework.
2.	Name two common interfaces in the Java Collections Framework.
3.	Explain the concept of autoboxing in Java in one sentence.
4.	What is the primary characteristic of a collection that implements the Map interface?
5.	In Java, which collection class is typically preferred for fast insertion and removal of elements in the middle of a collection: ArrayList or LinkedList?
6.	In a SortedSet, what determines the order of elements?
7.	What does JDBC stand for, and what is its primary purpose in Java?
8.	Briefly describe the purpose of the DAO (Data Access Object) pattern in Java database interactions.
9.	Explain the difference between ArrayList and LinkedList.
10.	Point out the purpose of the Iterator interface in Java collections.
11	Define JDBC in Java?
12	Compare the difference between HashSet and TreeSet.
13	Identify how do you sort elements in a collection in Java.
14	State the purpose of the Collections class in Java.
15	List some different types of JDBC drivers.
16	Point out common exceptions that can occur in JDBC.
17	Explain the concept of DAO.
18	What is the method used to sort the elements in a collection in Java.

19	Explain the concept of Map briefly.	
20	List out the ways to synchronize a collection in Java.	
	UNIT-V **PART – B**	
1.	Explain the Java Collections Framework, including the core interfaces and classes. Provide examples of situations where you would choose different collection types (e.g., List, Set, Map) based on the requirements of your program.	(13)
2.	Discuss the concept of autoboxing and unboxing in Java and how it simplifies the process of working with primitive data types in collections. Provide code examples that demonstrate autoboxing and unboxing in action.	(13)
3.	Using an Iterator is a common way to traverse elements in a collection. Explain the advantages and limitations of using an Iterator. Provide a code example that demonstrates how to use an Iterator to traverse a List.	(13)
4.	Java introduced the for-each loop as an alternative to using Iterators for collection traversal. Discuss the benefits of the for-each loop and provide code examples that showcase its usage with different types of collections.	(13)
5.	Explain the Collection interface in Java, including its key methods and characteristics. How does it differ from the List, Set, and Map interfaces? Provide code examples to illustrate the use of the Collection interface.	(13)
6.	Explain the concept of Map and its implementations in the Collections Framework.	(13)
7.	Compare and contrast the ArrayList and LinkedList classes in Java. Discuss the scenarios where one would be more suitable than the other. Provide code examples that demonstrate their differences in terms of performance and use cases.	(13)
8.	a) Write a Java code for reader writer problem. b) Describe how to implement Runnable interface for Linked List class.	(13)
9.	Describe about the common exceptions that can occur in JDBC and explain.	(13)
10.	Explain in detail about the concept of DAO.	(13)
11.	Describe the Set and SortedSet interfaces in Java. Provide examples of classes that implement these interfaces and explain how they ensure uniqueness and ordering of elements.	(13)
12.	Explain the differences between HashSet, LinkedHashSet, and TreeSet classes in Java. Provide scenarios where each of these classes would be the preferred choice for storing data.	(13)
13.	Discuss the process of storing user-defined classes in collections. What requirements must the user-defined class meet to work effectively within a collection? Provide a code example that demonstrates how to store and retrieve objects of a user-defined class in a collection.	(13)
14.	a) Identify the purpose of the Iterator interface in Java collections in detail. b) Develop a Java program that reads a collection of integers and uses the for-each loop to calculate their sum and average.	(13)
15.	Implement a Java program that uses a HashSet to store a list of unique email addresses. Ensure that the program can efficiently check for the presence of a specific email address in the HashSet.	(13)

PART – C		
1.	In a Java application, you are tasked with creating a system to manage student grades for different subjects. Each student is identified by a unique student ID (an integer), and you need to store the student's name and a map of subject names to corresponding grades. How would you use a Map to efficiently store the student information, mapping each student's ID to their name and a map of subjects to grades? Provide a code example demonstrating how you would create a Map to represent the student data structure. Include sample data for a few students, each with multiple subjects and corresponding grades. Explain how you would add a new student to the system, update the grades for a subject for an existing student, and retrieve the grades for a specific subject for a particular student using Map methods.	(15)
2.	Explain the purpose and usage of the TreeSet class in the Java Collection Framework. Provide an example of how you would create a TreeSet, add elements to it, and perform various operations such as sorting, iterating, and removing elements from the TreeSet. Also, discuss the characteristics and advantages of using a TreeSet over other collection classes	(15)
3.	Explain the concept of JDBC (Java Database Connectivity) and its role in connecting Java applications to databases. Discuss the advantages of using a DAO (Data Access Object) pattern for database interactions in Java. Provide a code example that demonstrates how to establish a database connection and perform CRUD (Create, Read, Update, Delete) operations using JDBC and a DAO.	(15)
4.	Explain the steps involved in establishing a database connection using JDBC.	(15)
5.	Build a Java program that demonstrates the use of the Collection interface to manage a list of student names. Implement methods for adding, removing, and displaying student names.	(15)

****