CSA-0943-Programming Java

**Operators and Controls:-**

1. Write a Java program to implement insertion sort on an array of integers using loops and conditional statements
2. Write a Java program to implement bubble sort on an array of integers using loops and conditional statements
3. Write a Java program to find the smallest element in an array of integers using loops and conditional statements
4. Write a Java program to find the second smallest element in an array of integers using loops and conditional statements
5. Write a Java program to find the largest element in an array of integers using loops and conditional statements.
6. Write a Java program to find the second largest element in an array of Integers using loops and conditional statements.
7. Write a Java program to print the following pattern using loops: 1 22 333 4444 55555.
8. Write a Java program to check whether a given number is a perfect number or not using loops and conditional statements.
9. Write a Java program to check whether a given number is an Amstrong number or not using loops and conditional statements.
10. Write a Java program to check whether a given string is a palindrome or not using loops and conditional statements
11. Write a Java program to calculate the volume of a sphere using the radius input by the user.
12. Write a Java program to find the power of a number using loops and conditional statements.
13. Write a Java program to calculate the area of a circle using the radius input by the user.
14. Write a Java program to calculate the product of the digits of a given number using loops and conditional statements.
15. Witte is a Java program to convert a binary number to a decimal using loops and conditional statements.
16. Write a Java program be calculate the sum of the digits of a given number using loops and conditional statements.
17. Write a Java program to convert a decimal number to binary using loops and conditional statements.
18. Write a Java program to find the LCM (Least Common Multiple) of two numbers using loops and conditional statements.
19. Write a Java program to find the factorial of a given number using recursion.
20. Write a Java program to print the following pattern using loops \*\*\*\*\* \*\*\*\* \*\*\* \*\* \*
21. Write a Java program to print the multiplication table of a given number using loops.
22. Write a Java program to check whether a given number is a prime number or not using loops and conditional statements.
23. Write a Java program to implement selection sort on an array of integers using loops and conditional statements.
24. Write a Java program to find the GCD (Greatest Common Divisor) of two numbers using loops and conditional statements.
25. Write a Java program to implement binary search on an array of integers using loops and conditional statements.
26. Write a Java program to print the Fibonacci series up to a given number using loops.
27. Write a Java program to calculate the factorial of a given number using loops.
28. Write a Java program to print the following pattern using loops: \* \*\* \*\*\* \*\*\*\* \*\*\*\*\*
29. Write a Java program to check whether a given year is a leap year or not using conditional statements.
30. Write a Java program that takes two integers as input from the user and calculates their sum, difference, product, and quotient using arithmetic operators.
31. Write a Java program to find the maximum of three numbers using conditional statements.

**OOPS CONCEPTS:-**

1. Write a Java program to create a class called Student with a constructor that takes in a name, ID number, and a list of grades, and methods to calculate and return the student's average grade and letter grade
2. Write a Java program to create a class called Car with a constructor that takes in the make, model, and year of the car, and a method to print out the car's make, model, and year.
3. Write a Java program to create a class called BankAccount with a constructor that takes in an account number and an initial balance, and methods to deposit and withdraw money from the account.
4. Write a Java program to create a class called Rectangle with a constructor that takes in the length and width of the rectangle, and a method to calculate and retum the area of the rectangle.
5. Write a Java program to create a class called Person with a constructor that takes in a name and age, and a method to print out the person's name and age. (Overloading).
6. Write a Java program to demonstrate method overriding with a simple calculator. Create a parent class called Calculator with methods such as add(). subtract(), multiply(), and divide(). Then, create a child class called Scientific Calculator that overrides the multiply() method to perform a more complex calculation.
7. Write a Java program to demonstrate method overloading with variable arguments. Create a method called sum() that takes in a variable number of integers and returns their sum. Then, create an overloaded version of the sum() method that takes in a variable number of doubles and returns their sum.
8. Write a Java program to demonstrate polymorphism with abstract classes. Create an abstract class called Animal with abstract methods such as eat() and sleep(). Then, create child classes such as Dog and Cat that inherit from the Animal class and implement their own versions of the eat() and sleep() methods.
9. Write a Java program to demonstrate polymorphism with interfaces. Create an interface called Drawable with a method called draw(). Then, create classes such as Circle and Square that implement the Drawable interface and have their own unique implementations of the draw() method.
10. Write a Java program to implement a basic shape hierarchy using polymorphism. Create a parent class called Shape with attributes such as area and perimeter. Then, create child classes such as Circle and Rectangle that inherit from the Shape class and have their own unique methods for calculating area and perimeter.
11. Write a Java program to implement a basic shape hierarchy using polymorphism. Create a parent class called Shape with attributes such as area and perimeter. Then, create child classes such as Circle and Rectangle that inherit from the Shape class and have their own unique methods for calculating area and perimeter.
12. . Write a Java program to implement a basic car rental system using inheritance. Create a parent class called Vehicle with attributes such as make and model. Then, create child classes such as Sedan and SUV that inherit from the Vehicle class and have their own unique attributes and methods for renting and returning cars.
13. Write a Java program to implement a basic weather forecasting system using inheritance. Create a parent class called Forecast with attributes such as date and temperature. Then, create child classes such as SunnyForecast and RainyForecast that inherit from the Forecast class and have their own unique attributes and methods for predicting the weather.
14. Write a Java program to implement a basic restaurant management system using Inheritance. Create a parent class called Menu with attributes such as name and price. Then, create child classes such as Appetizer and Entree that inherit from the Menu class and have their own unique attributes and methods for ordering and serving food.
15. . Write a Java program to implement a basic game system using inheritance. Create a parent class called Game with attributes such as title and genre. Then, create child classes such as ActionGame and PuzzleGame that inherit from the Game class and have their own unique attributes and methods for playing the game.
16. Write a Java program to implement a basic university management system using inheritance. Create a parent class called Person with attributes such as name and age. Then, create child classes such as Student and Professor that inherit from the Person class and have their own unique attributes and methods for managing coursework and teaching classes.
17. Write a Java program to implement a basic library management system using inheritance. Create a parent class called Item with attributes such as title and author. Then, create child classes such as Book and DVD that inherit from the Item class and have their own unique attributes and methods for borrowing and returning items.
18. . Write a Java program to implement a basic shape hierarchy using inheritance. Create a parent class called Shape with attributes such as color and area. Then, create child classes such as Circle, Square, and Triangle that inherit from the Shape class and have their own unique methods for calculating area and perimeter.
19. Write a Java program to implement a basic animal classification system using inheritance. Create a parent class called Animal with attributes such as name and habitat. Then, create child classes such as Mammal, Reptile, and Bird that inherit from the Animal class and have their own unique methods for eating and reproducing.
20. Write a Java program to implement a basic employee management system using inheritance. Create a parent class called Employee with attributes such as name, id, and salary. Then, create child classes such as Manager and Sales Person that inherit from the Employee class and have their own unique methods for managing teams and making sales.
21. Write a Java program to implement a basic banking system using inheritance. Create a parent class called Account with attributes such as account number, account holder name, and balance. Then, create child classes such as SavingsAccount and CheckingAccount that inherit from the Account class and have their own unique methods for withdrawing and depositing money. Sample Input: SavingsAccount Account Number: 123456789 Account Holder Name: Jane Doe Interest Rate (%): 1.5 Initial Deposit: 5000 CheckingAccount Account Number: 987654321 Account Holder Name: John Smith Overdraft Limit: 1000 Initial Deposit: 1000 Sample Output: The accounts in the system are: 1. Savings Account (123456789) - Jane Doe (Balance: $5000.00, Interest Rate: 1.5%) 2. Checking Account (987654321) - John Smith (Balance: $1000.00, Overdraft Limit: $1000.00) Please enter the number of the account you wish to perform a transaction on: 1 Please enter the transaction type (deposit/withdraw): deposit Please enter the amount to deposit: 2500 Transaction successful. New balance: $7500.00 Please enter the number of the account you wish to perform a transaction on: 2 Please enter the transaction type (deposit/withdraw): withdraw Please enter the amount to withdraw: 2000 Transaction successful. New balance: $-1000.00 (overdraft limit reached).

**EXCEPTION HANDLING:-**

1. Write a Java program to demonstrate how to use the "Join" method to wait for a thread to complete its execution before proceeding with the main thread.
2. Write a Java program to create a thread pool that can execute multiple threads concurrently.
3. Write a Java program to demonstrate how to use the "wait" and "notify" methods to implement a producer-consumer scenario.
4. Write a Java program to create a deadlock scenario using multiple threads.
5. Write a Java program to create a producer-consumer scenario using multiple threads. The producer thread should add items to a queue, and the consumer thread should remove items from the queue and process them.
6. Write a Java program to create multiple threads that perform some time-consuming task, such as calculating prime numbers or sorting an array, and compare their execution times.
7. . Write a Java program to create a thread-safe counter that can be accessed by multiple threads simultaneously.
8. Write a Java program to create a simple thread that prints the numbers from 1 to 10.

**JAVA COLLECTION AND FRAMEWORK:-**

1. . Write a Java program to create a Set of integers and perform different operations on it, such as finding the maximum and minimum values.
2. Write a Java program to create a LinkedHashMap of strings and integers and retrieve the keys in the order they were inserted.
3. Write a Java program to create a PriorityQueue of integers and add elements to it.
4. Write a Java program to create a TreeMap of strings and Integers and iterate over its entries.
5. Write a Java program to create a HashSet of strings and remove a specific element from it.
6. . Write a Java program to create a list of integers and sort them in ascending order using the Collections framework.
7. You are developing a game that involves different types of characters, such as heroes, villains, and monsters. Write a Java program to create an interface called "Character" with methods for moving, attacking, and taking damage. Implement the interface in classes for each type of character, such as "Hero", "Villain", and "Monster".
8. . You are working on a project that requires the use of a logging framework. Write a Java program to create an interface called "Logger" with methods for logging different types of messages, such as "debug", "info", "warning", and "error". Implement the interface in a class called "Log4jLogger" that uses the Log4j logging framework.

**Package, interface, and string handling:-**

1. Write a Java program to convert a given string to a char array.
2. Write a Java program to trim a given string and remove leading and trailing spaces.
3. Write a Java program to split a given string into an array of substrings based on a specified delimiter.
4. Write a Java program to replace a specified character with another character in a given string.
5. Write a Java program to remove a specified character from a given string.
6. Write a Java program to convert a given string to uppercase. Write a Java program to find the index of a given character or substring in a given string.
7. Write a Java program to check if a given string ends with a specific suffix.
8. Write a Java program to compare two strings lexicographically.
9. Write a Java program to create a substring from a given string.
10. Write a Java program to find the most frequent word in a given string.
11. Write a Java program to find the first non-repeated character in a given string.
12. Write a Java program to concatenate two given strings.
13. Write a Java program to reverse a given string.
14. Write a Java program to remove all white spaces from a given string.
15. Write a Java program to convert a given string to lowercase.
16. Write a Java program to find the length of the longest substring without repeating characters in a given string.
17. Write a Java program to check if a given string contains only digits.
18. Write a Java program to count the number of occurrences of a given character in a string.
19. Write a Java program to check if a given string is a palindrome or not.

**Synchronization and Interthread:-**

1.Write a Java program to demonstrate how to use the AtomicInteger class to provide atomic access to a shared integer variable.

Output:

Shared value: 20000

This confirms that the two threads were able to increment the shared value atomically and the final value of the shared value i.

2. Write a Java program to demonstrate how to use synchronization to prevent multiple threads from accessing a shared resource simultaneously.