Q1.Java method overloading implements the OOPS concept
A. Encapsulation
B. Inheritance
C. Polymorphism
D. Abstraction
Ans. C. Polymorphism
Java method overloading is a form of polymorphism where a class can have multiple methods with the same name but different parameter lists. This allows you to perform different operations using the same method name, making code more readable and easier to use.

Q2.Data members and member functions of a class are private by default.
A. True
B. False
C. Depend on code
D. None
Ans. A. True
In Java, data members (fields) and member functions (methods) of a class are private by default. This means that they can only be accessed within the same class and not from outside. If you want to allow access from outside the class, you need to explicitly specify access modifiers such as public, protected, or package-private (default).

Q3.Which of the following functions can be inherited from the base class?
A. Constructor
B. Static
C. All
D. None
Ans. B. Static
Static functions in a base class cannot be inherited by derived classes. They are associated with the class itself, rather than instances of the class, so they are not part of the inheritance hierarchy. However, they can be accessed using the class name itself, without the need for an instance of the class.

Q4. Identify the feature, which is used to reduce the use of nested classes.
A. Binding
B. Abstraction
C. Inheritance
D. None
Ans. B. Abstraction
Abstraction is the feature that is used to reduce the use of nested classes. Abstraction allows you to hide the complex implementation details and show only the necessary features or

functionalities to the user. This can help in avoiding the need for deeply nested classes and making the code more manageable and understandable.

Q5. Which concept of Java is achieved by combining methods and attributes into a class?
A. Encapsulation
B. Inheritance
C. Polymorphism
D. Abstraction
Ans. A. Encapsulation
The concept of encapsulation in Java is achieved by combining methods and attributes (data members) into a class. Encapsulation refers to the practice of bundling data (attributes) and the methods (functions) that operate on that data into a single unit, which is the class in this context. This helps in controlling access to the data and providing a clear interface for interacting with the object's internal state.

Q6.Which of the following declarations does not compile?
A. double num1, int num2 = 0;
B. int num1, num2;
C. int num1, num2 = 0;
D. int num1 = 0, num2 = 0;
Ans. A. double num1, int num2 = 0;
In Java, you cannot declare variables of different types in a single declaration like this. Each variable declaration must specify only one data type. The correct way to declare variables of different types is to declare them separately, like in option D:
D. int num1 = 0, num2 = 0;

Q7.Which of these interface must contain a unique element?
A. Set
B. List
C. Array
D. collection
Ans. A. Set
In Java, an interface is a reference type that is similar to a class. It is a collection of abstract methods. The Set interface is a part of the Java Collections Framework and represents a collection of elements with no duplicate values. Since a set, by definition, cannot contain duplicate elements, the Set interface enforces uniqueness among its elements. Therefore, any class that implements the Set interface must ensure that it does not allow duplicate elements.

Q8.Predict the output? package main; class T { int t = 20; } class Main { public static void main(String args[]) { T t1 = new T(); System.out.println(t1.t); } }

A. 20
B. 0
C. COMPILE ERROR
Ans. A. 20
The output will be 20. In the given code, an object of class T is created using new T(). The constructor of class T is not explicitly defined, so the default constructor is used. Since t is an instance variable with a value of 20, when the object t1 is created, its t attribute is initialized to 20. When System.out.println(t1.t) is executed, it prints the value of t from the t1 object, which is 20.

Q9. What is the output of the below Java program? //bingo.java file public class Hello { public static void main(String[] args) { System.out.println("BINGO"); } }
A. BINGO
B. bingo
C. 0
D. Compile Error
Ans. A. BINGO
The output of the given Java program will be "BINGO". The program defines a class named Hello with a main method. Inside the main method, the System.out.println("BINGO"); statement is used to print "BINGO" to the console. Since the letter case is preserved in the string "BINGO," the output will exactly match the string and will be displayed as "BINGO" in the console.

Q10.What will be the output of the following Java program? class variable_scope { public static void main(String args[]) { int x; x = 5; { int y = 6; System.out.print(x + " " + y); } System.out.println(x + " " + y); } }
A. Compilation Error
B. Runtime Error
C. 5 6 5 6
D. 5 6 5
Ans. A. Compilation Error
The code will not compile because of the variable scope issue. In the inner block, you're declaring a variable y, but you're trying to access it outside of that block. This will result in a compilation error since the variable y is not recognized in the outer scope of the block.
So, the correct answer is A. Compilation Error.

Q11.What will be the output of the following Java code? class String_demo { public static void main(String args[]) { char chars[] = {'a', 'b', 'c'}; String s = new String(chars); System.out.println(s); } }
A. abc
B. a
C. b

D. c
Ans. A. abc
The output of the given Java code will be "abc". In the code, an array of characters chars is created with the values 'a', 'b', and 'c'. Then a new String object s is created using the constructor that takes an array of characters. This constructor constructs a new string with the characters from the provided array.
When System.out.println(s); is executed, it will print the content of the string s, which is "abc".

Q12. What will be the output of the following Java program? final class A { int i; } class B extends A { int j; System.out.println(j + " " + i); } class inheritance { public static void main(String args[]) { B obj = new B(); obj.display(); } } WORKSHEET
A. 2 2
B. 3 3
C. Runtime Error
D. Compilation Error
Ans. D. Compilation Error

The given code has several issues that will prevent it from compiling:

The class B is attempting to extend the final class A, which is not allowed. Final classes cannot be extended.
The code that tries to print j and i is directly inside the class B, outside of any method. Statements in Java should be inside methods, constructors, or static initializers.
The method display() is mentioned in the main method, but it is not defined in the class B.
Due to these compilation errors, the correct answer is D. Compilation Error.

Q13.What is output of following program public class Test { public int getData() //getdata() 1 { return 0; } public long getData() //getdata 2 { return 1; } public static void main(String[] args) { Test obj = new Test(); System.out.println(obj.getData()); } }
A. 1
B. 0
C. Runtime Error
D. Compilation Error
Ans. D. Compilation Error

In Java, you cannot have two methods in the same class with the same name and parameter list but different return types. This would lead to ambiguity when trying to call the method. In the given code, the methods getData() have the same name and parameter list, but they have different return types (int and long).

Due to this ambiguity, the code will not compile, and you will get a compilation error.

Q14. What is the output of the following program? public class Test{ static int start = 2; final int end; public Test(int x) { x = 4; end = x; } public void fly(int distance) { System.out.println(end-start+" "); System.out.println(distance); } public static void main(String []args){ new Test(10).fly(5); } }
A. [2 5]
B. [0 0]
C. [5 2]
D. [0 2]
Ans. D. [0 2]

In this code, a new Test object is created using the constructor with an argument of 10. Inside the constructor, x is assigned the value 4, but this value is not used further. Instead, the end field is assigned the value of x, which is 4.

Then, the fly method is called on the newly created object with an argument of 5. Inside the fly method, it prints the result of end - start, which is 4 - 2 = 2, followed by the value of distance, which is 5.

So, the output will be: 2 5

Q15.What is the output of the following program? String john = "john"; String jon = new String(john); System.out.println((john==jon) + " "+ (john.equals(jon)));
A. true true
B. true false
C. false true
D. false false
Ans. C. false true

Explanation:

john is assigned the string literal "john".
jon is assigned a new String object created with the content of john.
(john==jon) compares the references of john and jon, which will be false because jon is a new String object, not the same reference as john.
(john.equals(jon)) compares the contents of john and jon, which will be true because the contents of the two strings are the same ("john").
So, the output will be "false true".

Q16. Given that Student is a class, how many reference variables and objects are created by the following code? Student studentName, studentId; studentName = new Student(); Student stud_class = new Student();

A. Three reference variables and two objects are created.
B. Two reference variables and two objects are created.
C. One reference variable and two objects are created.
D. Three reference variables and three objects are created.
Ans. B. Two reference variables and two objects are created.

Explanation:

Student studentName, studentId; declares two reference variables studentName and studentId.
studentName = new Student(); creates a new Student object and assigns its reference to the studentName variable.
Student stud_class = new Student(); creates another new Student object and assigns its reference to the stud_class variable.
So, there are two reference variables (studentName and stud_class) and two objects created in total.

Q17. Write a java program to check even or odd number?

Ans. import java.util.Scanner;

```
public class EvenOddChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        if (number % 2 == 0) {
            System.out.println(number + " is an even number.");
        } else {
            System.out.println(number + " is an odd number.");
        }

        scanner.close();
    }
}
```

Q18. Write a java program to find average of two numbers?

Ans. import java.util.Scanner;

```
public class AverageCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        double number1 = scanner.nextDouble();

        System.out.print("Enter the second number: ");
        double number2 = scanner.nextDouble();

        double average = (number1 + number2) / 2;

        System.out.println("The average of " + number1 + " and " + number2 + " is: " + average);

        scanner.close();
    }
}
```

Q19. Write a java program to swap two numbers?

Ans. import java.util.Scanner;

```
public class NumberSwapper {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        int number1 = scanner.nextInt();

        System.out.print("Enter the second number: ");
        int number2 = scanner.nextInt();

        System.out.println("Before swapping: Number1 = " + number1 + ", Number2 = " +
number2);

        // Swapping using a temporary variable
        int temp = number1;
        number1 = number2;
        number2 = temp;

        System.out.println("After swapping: Number1 = " + number1 + ", Number2 = " + number2);

        scanner.close();
    }
}
```

Q20. Write a java program to check whether a number is prime or not?

Ans. import java.util.Scanner;

```
public class PrimeNumberChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        if (isPrime(number)) {
            System.out.println(number + " is a prime number.");
        } else {
            System.out.println(number + " is not a prime number.");
        }

        scanner.close();
    }

    public static boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }

        // Check for divisibility from 2 to the square root of the number
        for (int i = 2; i * i <= num; i++) {
            if (num % i == 0) {
                return false; // Found a divisor other than 1 and itself
            }
        }

        return true; // No divisors other than 1 and itself
    }
}
```

Q21. Write a java program to find table of n?

Ans. import java.util.Scanner;

```
public class MultiplicationTable {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the number for the table: ");
        int number = scanner.nextInt();

        System.out.print("Enter the limit for the table: ");
        int limit = scanner.nextInt();

        System.out.println("Multiplication table of " + number + " up to " + limit + ":");
        for (int i = 1; i <= limit; i++) {
            System.out.println(number + " x " + i + " = " + (number * i));
        }

        scanner.close();
    }
}
```

Q22. Write a java program to find the largest of three numbers?

Ans. import java.util.Scanner;

```
public class LargestOfThree {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the first number: ");
        double number1 = scanner.nextDouble();

        System.out.print("Enter the second number: ");
        double number2 = scanner.nextDouble();

        System.out.print("Enter the third number: ");
        double number3 = scanner.nextDouble();

        double largest = Math.max(number1, Math.max(number2, number3));

        System.out.println("The largest number is: " + largest);

        scanner.close();
    }
}
```

Q23. Write a java program to calculate Simple Interest?
Ans. import java.util.Scanner;

```
public class SimpleInterestCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the principal amount: ");
        double principal = scanner.nextDouble();

        System.out.print("Enter the rate of interest (in percentage): ");
        double rate = scanner.nextDouble();

        System.out.print("Enter the time period (in years): ");
        double time = scanner.nextDouble();

        double simpleInterest = (principal * rate * time) / 100;

        System.out.println("Simple Interest: " + simpleInterest);

        scanner.close();
    }
}
```

Q24. Write a java program to calculate Area and perimeter of Rectangle?
Ans. import java.util.Scanner;

```
public class RectangleCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the length of the rectangle: ");
        double length = scanner.nextDouble();

        System.out.print("Enter the width of the rectangle: ");
        double width = scanner.nextDouble();

        double area = length * width;
        double perimeter = 2 * (length + width);
        System.out.println("Area of the rectangle: " + area);
        System.out.println("Perimeter of the rectangle: " + perimeter);

        scanner.close();
    }
}
```

Q25. Write a java program to check whether character is vowel or consonant?

Ans. import java.util.Scanner;

```
public class VowelConsonantChecker {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a character: ");
        char ch = scanner.next().charAt(0);

        if (isVowel(ch)) {
            System.out.println(ch + " is a vowel.");
        } else {
            System.out.println(ch + " is a consonant.");
        }

        scanner.close();
    }

    public static boolean isVowel(char ch) {
        ch = Character.toLowerCase(ch);
        return ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u';
    }
}
```