JAVA PROGRAMMING ASSIGNMENT – 4

PART A – THEORY QUESTIONS

1)What is abstraction in Java?

➔We know what object do but don't know how.

-Data Hiding.

-Using abstract class and interface we can acheive.

-In abstraction we are hide a logic in between method.

-'abstract' keyword is used.

-hiding implementation details and showing only what is necessary.

-Interface supports 100% abstraction.

2)What is an abstract class?

Ans:-We cannot create objects of abstract classes.

➔we use abstract keyword to declare an abstract class.

-an abstract class can have both the regular methods and abstract methods.

-a method that doesn't have its body is known as an abstract method.

-we can access members of the abstract class using the object of the subclass.

-It abstract class includes any abstract methods,then all the child classes inherited from the abstract superclass much provide the implementation of abstract method.

Q3)What is an interface in Java?

➔using 'interface' keyword we can vreate interface.

-Interface is a contract between class and methods.

-Interface can contain static ,default and abstract method.

-by default any method is public abstract.

-By default any data member is public static final

-we can extends inetrface with other interface using 'extends' keyword .

-using inetrface we can acheive multiple inheritance,abstraction in java.

-'implement' keyword is used for override methods from interface.

-nested interface is also allowed

-types=1.functional

2.mutable

3.simple

-how to implements:-1.subclass

2.Anonymous inner class

3.Lambda

<u>4)Difference between abstract class and interface.</u>

➔ Abstract Class

1.Declared using the abstract keyword.

2.Can contain abstract and non-abstract methods.

3.Can have instance variables, static variables, and final variables.

4.Constructors are allowed.

5.Methods can use private, protected, default, and public access modifiers.

6.Supports single inheritance only.

7.Uses the extends keyword for inheritance.

8.Object of abstract class cannot be created directly.

9.Provides code reusability.

10.Faster than interface.

Interface

1.Declared using the interface keyword.

2.All methods are abstract by default.

3.Variables are public static final by default.

4.Constructors are not allowed.

5.Methods are public by default.

6.Supports multiple inheritance.

7.Uses the implements keyword for inheritance.

8.Object of interface cannot be created directly.

9.Used to achieve 100% abstraction (before Java 8).

10.Provides only behavior, not implementation.

11.Slightly slower than abstract class.

<u>5)What is a constructor?</u>

➔1.A constructor has the same name as the class.

2.It does not have a return type, not even void.

3.It is called automatically when an object is created using new.

4.Used to initialize data members of a class.

5.A constructor can be public, protected, or default (not private for object creation).

6.Constructors can be overloaded.

7.A class can have more than one constructor.

8.If no constructor is written, Java provides a default constructor.

9.Constructors are not inherited.

10.Constructors cannot be static, abstract, or final.

PART B – PROGRAMMING QUESTIONS

<u>PART B – PROGRAMMING QUESTIONS</u>

<u>1)Write a program to count the number of digits in a number.</u>

➔import java.util.Scanner;

```
class CountDigits {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter a number: ");
 int n = sc.nextInt();
 int count = 0;
 if (n == 0) {
 count = 1;
 } else {
 while (n != 0) {
 count++;
 n = n / 10;
 }
 }
 System.out.println("Number of digits: " + count);
 }
}
```

<u>2)Write a program to find the greatest common divisor (GCD).</u>

➔import java.util.Scanner;

```
class GCD {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter first number: ");
 int a = sc.nextInt();
 System.out.print("Enter second number: ");
 int b = sc.nextInt();
 while (b != 0) {
 int temp = b;
 b = a % b;
 a = temp;
 }
 System.out.println("GCD is: " + a);
 }
}
```

3)Write a program to calculate LCM of two numbers.

```
➔import java.util.Scanner;
class LCM {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter first number: ");
 int a = sc.nextInt();
 System.out.print("Enter second number: ");
 int b = sc.nextInt();
 int x = a, y = b;
 while (a != b) {
 if (a > b)
 a = a - b;
 else
 b = b - a;
 }
 int gcd = a;
 int lcm = (x * y) / gcd;
 System.out.println("LCM is: " + lcm);
 }
}
```

4)Write a program to check if a year is leap year.

```
➔import java.util.Scanner;
class LeapYear {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter a year: ");
 int year = sc.nextInt();
 if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
 System.out.println("Leap Year");
 } else {
 System.out.println("Not a Leap Year");
 }
 }
}
```

5)Write a program to print all even numbers between 1 and 50.

```
➔import java.util.Scanner;
class EvenNumbers {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter the limit: ");
 int n = sc.nextInt();
 for (int i = 1; i <= n; i++) {
 if (i % 2 == 0) {
 System.out.print(i + " ");
 }
 }
 }
}
```

6)Write a program to calculate power of a number.

```
➔import java.util.Scanner;
class Power {
 public static void main(String[] args) {
 Scanner sc = new Scanner(System.in);
 System.out.print("Enter base: ");
 int base = sc.nextInt();
 System.out.print("Enter exponent: ");
 int exp = sc.nextInt();
 int result = 1;
 for (int i = 1; i <= exp; i++) {
 result = result * base;
 }
 System.out.println("Power is: " + result);
 }
}
```