JAVA PROGRAMMING ASSIGNMENT – 5

PART A – THEORY QUESTIONS

1)**Difference between default,parameterized ,and copy constructors.**

Ans:

Feature	**Default Constructor**	**Parameterized Constructor**	**Copy Constructor**
Parameters	No	Yes	Object reference
Purpose	Initialize with default values	Initialize with given values	Copy values from another object
Provided by Java	Yes (if no constructor exists)	No	No
Flexibility	Least	More	Used for cloning data
Example Call	new Student()	new Student(1,"Asha")	new Student(s1)

2)**What is the use of this keyword ?**

Ans: the this keyword is used to **refer to the current object** (the object that is calling the method or constructor).

Think of this as: **“this current object”**

Main uses of this keyword:

1)To differentiate instance variables and local variables

```
class Student {
   int id;
   Student(int id) {
      this.id = id;   // this.id = instance variable
   }
}
```

Without this, Java gets confused between local variable and instance variable.

2)To call current class method

```
class Demo {
   void show() {
      System.out.println("Show method");
   }
   void display() {
      this.show();   // calling current class method
   }
}
```

3)To call current class constructor.

4)To return current object.

5)this keyword is used to refer to the current object, call current class methods/constructors, and differentiate instance and local variables

3) **What is the use of super keyword ?**

Ans: **super** refers to the parent (superclass) object.

1)Access parent class variable.

When parent and child have same variable name.

```
class A {
   int x = 10;
}
class B extends A {
   int x = 20;
   void show() {
      System.out.println(super.x); // parent variable
   }
```

}

2)Call parent class method.

When child overrides parent method.

```
class A {
   void display() {
      System.out.println("Parent method");
   }
}
class B extends A {
   void display() {
      super.display(); // calls parent method
      System.out.println("Child method");
   }
}
```

3)super keyword is used to refer to the parent class object, access parent members, and call parent constructors.

4**) What is the use of static keyword ?**

Ans: static means the member boleng to class,not to object ,not to objects.

Where static is used.

1)static variables 2)static method

3)static block 4)static nested class

1)static variables

1)shared by all objects.

2)only one copy in memory.

2)static methods

1)called using class name.

2)cannot use non-static variables directly.

3)static block

1)executes when class is loaded.

2)used to initialize static data.

Key Points:

1)static members belong to class.

2)Access using className.member.

3)static member cannot use this.

4)main() method is static

5)**What are static blocks and static methods?**

Ans**: Static Block in Java**

A static block is a block of code that is executed only once when the class is loaded into memory.

It runs before the main() method.

Why do we use Static Block?

To perform one-time initialization

To initialize static variables

To load drivers (example: database drivers)

Executes only once

Runs before main()

Can have multiple static blocks (executed in order)

Cannot use this or super

Used only for static data

Static Methods

A static method belongs to the class, not to objects.

It can be called without creating an object.

Why do we use Static Method?

When logic does not depend on object data

Utility methods (e.g. Math.max())

Called directly using class name

Can access only static variables and methods

Cannot use this and super

Cannot access instance variables directly

PART B – PROGRAMMING QUESTIONS

1)**Programs t find the sum of n natural numbers**.

Ans:
```
import java.util.Scanner;
  class SumNatural {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter n: ");
      int n = sc.nextInt();
      int sum = 0;
      for (int i = 1; i <= n; i++) {
         sum = sum + i;
      }
      System.out.println("Sum = " + sum);
   }
}
```

2)**Write a program to reverse string.**

Ans:
```
import java.util.Scanner;
class ReverseString {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      String str, rev = "";
      System.out.print("Enter String: ");
      str = sc.nextLine();
      for(int i = str.length() - 1; i >= 0; i--) {
         rev += str.charAt(i);
      }
      System.out.println("Reversed: " + rev);
   }
}
```

3)**Write a program to check if a String is palindrome.**

Ans:
```
class Palindrome {
   public static void main(String[] args) {
      String str = "madam";
      String rev = "";
      for(int i = str.length() - 1; i >= 0; i--) {
         rev += str.charAt(i);
      }
      if(str.equals(rev))
         System.out.println("Palindrome");
      else
         System.out.println("Not Palindrome");
```

```
    }
}
```

4**)Write a program to count vowels and consonants in a String**

Ans:

```
import java.util.Scanner;
class VowelConsonantCount {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String s = sc.nextLine().toLowerCase();
        int vowels = 0, consonants = 0;
        for (int i = 0; i < s.length(); i++) {
            char ch = s.charAt(i);
            if (ch >= 'a' && ch <= 'z') {
                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u') {
                    vowels++;
                } else {
                    consonants++;
                }
            }
        }

        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
    }
}
```

5)**Write a program to count words in a sentence.**

Ans:
```
import java.util.Scanner;
class WordCount {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a sentence: ");
        String s = sc.nextLine();
        int count = 1;
        for (int i = 0; i < s.length(); i++) {
            if (s.charAt(i) == ' ') {
                count++;
            }
        }
        System.out.println("Number of words: " + count);
    }
}
```