Date:29/may/2024- Assignment

1. What is protected access modifier and write a program using protected?
   Answer:

```
class Parent {
    protected void display() {
        System.out.println("This is a protected method.");
    }
}

class Child extends Parent {
    public static void main(String[] args) {
        Child obj = new Child();
        obj.display();
    }
}
```

2. What is the method overloding and write a program using overloading?
   Answer:

```
class MethodOverloading {

    private static void display(int a) {

        System.out.println("Got Integer data.");

    }


    private static void display(String a) {

        System.out.println("Got String object.");

    }


    public static void main(String[] args) {

        display(1);       // Output: Got Integer data.

        display("Hello"); // Output: Got String object.

    }

}
```

3.what is the output of the below program

```
class MethodOverloading {
    private static void display(int a) {
        System.out.println("Arguments: " + a);
    }

    private static void display(int a, int b) {
        System.out.println("Arguments: " + a + " and " + b);
    }

    public static void main(String[] args) {
```

```
        display(1);
        display(1, 4);
    }
}Arguments: 1
Arguments: 1 and 4
```

This output is produced by calling the `display` method with different sets of arguments. The first call to `display(1)` calls the method with a single integer parameter, which prints "Arguments: 1". The second call to `display(1, 4)` calls the overloaded method with two integer parameters, which prints "Arguments: 1 and 4".

4.what is the output of the below program?

```
class A {
    String s = "Class A";
}

class B extends A {
    String s = "Class B";

    { System.out.println(super.s); }
}

class C extends B {
    String s = "Class C";

    { System.out.println(super.s); }
}

public class MainClass {
    public static void main(String[] args) {
        C c = new C();  // Output: Class A Class B
        System.out.println(c.s); // Output: Class C
    }
}
```