

Session-1

1. Print default values of static & instance variables for different data types.

Code:

```
public class DefaultValues {

    static int age1 = 23;
    static String name1 = "Sweety";
    static char a1 = 'j';
    static double b1 = 23.00;
    static float c1 = 12.56f;
    static boolean f1= false;

    int age = 23;
    String name = "Sweety";
    char a = 'j';
    double b = 23.00;
    float c = 12.56f;
    boolean f= false;

    public static void main(String args[]) {
        System.out.println("Static Datatypes Values : ");
        System.out.print(DefaultValues.age1+" ");
        System.out.print(DefaultValues.name1+" ");
        System.out.print(DefaultValues.a1+" ");
        System.out.print(DefaultValues.b1+" ");
        System.out.print(DefaultValues.c1+" ");
        System.out.print(DefaultValues.f1+" ");

        System.out.println();
        System.out.println("Instance Datatypes Values : ");
        DefaultValues obj = new DefaultValues();
        System.out.print(obj.age+" ");
        System.out.print(obj.name+" ");
        System.out.print(obj.a+" ");
        System.out.print(obj.b+" ");
        System.out.print(obj.c+" ");
    }
}
```

```
        System.out.print(obj.f+" ");  
    }  
}
```

Output:

```
Static Datatypes Values :  
23 Sweety j 23.0 12.56 false  
Instance Datatypes Values :  
23 Sweety j 23.0 12.56 false
```

2. Build a class Employee which contains details about the employee and compile and run its instance.

Code:

```
public class Employee {  
  
    String empName = "xyz";  
    int empid = 123;  
    double salary = 25000.00;  
  
    public static void main(String[] args) {  
        Employee emp = new Employee();  
        System.out.println("Employee name: " +emp.empName);  
        System.out.println("Employee id: "+emp.empid);  
        System.out.println("Employee salary: "+emp.salary);  
    }  
}
```

Output:

```
Employee name: xyz  
Employee id: 123  
Employee salary: 25000.0
```

3. Build a class that has references to other classes. Instantiate these reference variables and invoke instance methods.

Code:

```
public class CallByRef {  
  
    int val=78;  
  
    void modify(CallByRef obj) {  
  
        obj.val=obj.val+100;  
  
        System.out.println("In method:"+obj.val);  
    }  
    public static void main(String[] args) {  
  
        CallByRef obj= new CallByRef();  
        System.out.println("Before"+obj.val);  
  
        obj.modify(obj);  
        System.out.println("After"+obj.val);  
  
    }  
  
}
```

Output:

```
Before78  
In method:178  
After178
```

4. Create Student class with global variables and related methods with constructor and without constructor.

Code:

```

public class Student {

    static String student = " sj";
    static int stdid = 123;
    char department = 'a';
    double code = 2.00;

    static void StudentDetails() {
        System.out.println("Student Name:" +student);
        System.out.println("Student ID:" +stdid);
    }

    void Details() {
        System.out.println("Student Department:"
+department);
        System.out.println("Student code:" +code);}

    public static void main(String[] args) {
        Student.StudentDetails();
        Student obj = new Student();
        obj.Details();

    }

}

```

Output:

```

Student Name: sj
Student ID:123
Student Department:a
Student code:2.0

```

5. observe the difference between methods calling and constructor calling.

Code:

```
public class Calling {  
  
    static void displayMessage() {  
        System.out.println("Dsiplay Message");  
    }  
  
    void showMessage() {  
        System.out.println("Show Message");  
    }  
  
    public static void main(String[] args) {  
        displayMessage();  
        Calling obj = new Calling();  
        obj.showMessage();  
    }  
}
```

Output:

Dsiplay Message

Show Message