Assignment 1

- SARAVANAN

SCENARIO 1:

Committing mistakes, creating errors and learning :

1) class First
2) { public static void main(String[] args)
3) {
4) System.out.println("mistakes")
5) }
6) }

S.No	INPUTS	OUTPUTS
1)	Capital letter on class first letter, class First { …….. }	primary.java:1: error: class, interface, or enum expected Class First
2)	Removed brace at main method	primary.java:2: error: ')' expected { public static void main(String[] args ^
3)	Removed “ ; ” at line 4	primary.java:4: error: ';' expected System.out.println("mistakes") ^
4)	Removed small “p” and Replaced CAPITAL “ P ” at line 2 in main method.	primary.java:2: error: <identifier> expected { Public static void main(String[] args ^
5)	Removd “[ ]” on main method replaced “( )”	primary.java:2: error: <identifier> expected { public static void main(String() args)

		^ primary.java:2: error: ';' expected { public static void main(String() args) ^ primary.java:2: error: <identifier> expected { public static void main(String() args) ^
6)	Removed CAPITAL “S” from “System...” and Replaced small “ s ” at line 4 in printing statement.	primary.java:4: error: package system does not exist system.out.println("mistakes");
7)	Removed “args” in main method	primary.java:2: error: <identifier> expected { public static void main(String[]) ^
8)	Created non-similar class name to object name .	primary.java:5: error: cannot find symbol hello hl = new hello(); ^ symbol: class hello location: class First primary.java:5: error: cannot find symbol hello hl = new hello(); ^ symbol: class hello location: class First 2 errors
9)	Removed “{” or “}”	error: reached end of file while parsing } ^
10)	Running without save and without compile the file.	No changes from previous Outputs, OUTPUTS are same.

SCENARIO 2 :

Learning void, return statement, Method Calling, Method Definition :

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


         Demo demo = new Demo();
        int result = demo.add(10,20);
        int result1 = demo.multiply(5,2);
        System.out.println(result);
        System.out.println("multiplied answer is "+result1);
       }
        int add(int no1, int no2)
        {
         int result = no1 + no2;
        return result;
            }
            int multiply(int no1, int no2)
            {
            int result1 = no1 * no2;
      return result1;
      }
}
```

OUTPUTS:

“30”

“multiplied answer is 10”

SCENARIO 3 :

Method Calling, private :

```
public class School
{  int mark;

private int salary;
  static String school_name = "St. Antony's Primary School";

  void conduct_exams()
  {
        System.out.println("public exam");
  }
  void publish_results(int mark)
  {
  System.out.println(mark);
 }
}
```

```
public class Teacher {
        public static void main(String[] args) {
                School teacher = new School();
                teacher.conduct_exams();

                teacher.publish_results(75);

System.out.println(teacher.school_name);

//System.out.println(teacher.salary);  // ERROR SPOTTED..
```

The Error is “`The field School.salary is not visible`”.

```
}
}
```

OUTPUTS:

“public exam”

“75”

“St. Antony's Primary School”

SCENARIO 4 :

Learning private, default and public Access Modifiers, Creating Package and understanding its usage, Calling Methods with/without arguments.

package bank.chennai;

```
public class SBI {
	String empName,empId;
	public static String  branch_name = "chennai";

	public void get_loan(int amount){
		System.out.println(amount);
		return;
	}
	public void create_account() {
		return;
	}
	}
```

package bank.chennai;

```
public class Account_Holder {
public static void main(String[] args) {
		SBI acc = new SBI();
		acc.create_account();
```

```
            acc.get_loan(100);
            acc.empId = "15";
            acc.empName ="join";

        System.out.println(acc.empName);
        System.out.println(acc.empId);
        System.out.println(acc.branch_name);
        }
}
```

package bank.madurai;

```
import bank.chennai.SBI;

public class Account_Holder_Madurai extends SBI{
public static void main(String[] args) {
        SBI a1c = new SBI();
        a1c.create_account();
         a1c.get_loan(4000);
        System.out.println(a1c.branch_name);
}
}
```

SCENARIO 5 :

Understanding Multilevel Inheritance, Abstraction :

package assignment;

```
public abstract class HeadOffice {
	public void check_accounts(int amount) {
		System.out.println("money");
	}
	public int pay_tax(int amount) {
		return amount;
	}
	public abstract void receive_Customers();
}
```

package assignment;

```
public abstract class Branch_Plan extends HeadOffice {
	public static void main(String[] args) {
    System.out.println("office room");
	}
	public void do_interview() {
		System.out.println("Hello java");
	}
```

```
}
```

package assignment;

```
public class Branch extends Branch_Plan {
	public  void receive_Customers() {
		System.out.println("payment");
	}
	public static void main(String[] args) {
		Branch branch = new Branch();
		branch.do_interview();
		branch.check_accounts(1000);
		branch.pay_tax(2000);
		}
}
```