FULL STACK DEVELOPMENT – WORKSHEET 5

FIND OUTPUT OF THE PROGRAMS WITH EXPLANATION

Q1.//Stringbuffer

```
public class Main
{
	public static void main(String args[])
	{
		String s1 = "abc";
		String s2 = s1;
		s1 += "d";
		System.out.println(s1 + " " + s2 + " " + (s1 == s2));
		StringBuffer sb1 = new StringBuffer("abc");
		StringBuffer sb2 = sb1;
		sb1.append("d");
		System.out.println(sb1 + " " + sb2 + " " + (sb1 == sb2));
	}
}
```

Output : abcd abc false (Explanation : Here String s1 = abc is defined and s1 is assigned to s2. So both s1 and s2 refers to same contents i.e “abc”. After that s1 is concataneted using += operator and s1 refers to “abcd” while s2 still refers to the string “abc”, hence s1== s2 is false.)
abcd abcd true (Explanation : Here StringBuffer sb1 = “abc”, sb1 is assigned to sb2. So both sb1 and sb2 refering to same object. After sb1 is appended, its contents become “abcd”. As both sb1 and sb2 still refering to same object hence sb1==sb2 is true.)

Q2.// Method overloading

```
public class Main
{
	public static void FlipRobo(String s)
	{
		System.out.println("String");
	}
	public static void FlipRobo(Object o)
	{
		System.out.println("Object");
	}

	public static void main(String args[])
	{
		FlipRobo(null);
	}
}
```

Output : String (Explanation : Here FlipRobo method is overloaded. So when ‘FlipRobo(null)’ is called java compiler chooses most specific method with parameter type. So method with String type parameter is most suitable to argument null)

Q3.

```
class First
{
   public First() {  System.out.println("a"); }
}

class Second extends First
{
   public Second()  {  System.out.println("b"); }
}

class Third extends Second
{
   public Third()   {  System.out.println("c"); }
}
public class MainClass
{
   public static void main(String[] args)
   {
      Third c = new Third();
   }
}
```

Output : a
b
c

(Explanation : class Third object is created and constructor Third() is called. Third extends class Second so it invokes constructor Second, Second extends class First so constructor First is invoked. Hence first 'a' is printed then 'b' and then 'c')

Q4.

```
public class Calculator
{
   int num = 100;
   public void calc(int num)  { this.num = num * 10;  }
   public void printNum()     { System.out.println(num); }

   public static void main(String[] args)
   {
      Calculator obj = new Calculator();
      obj.calc(2);
      obj.printNum();
   }
}
```

Output : 20 (Explanation : in main method class Calcultor object is created. Respectively both calc() and printNum() methods are called. In calc method argument passed is '2'. In calc method "this.num=2*20" is performed. Hence value of num is 20. So PrintNum prints 20.)

Q5.

```
public class Test
{
   public static void main(String[] args)
   {
      StringBuilder s1 = new StringBuilder("Java");
      String s2 = "Love";
      s1.append(s2);
      s1.substring(4);
      int foundAt = s1.indexOf(s2);
      System.out.println(foundAt);
   }
}
```

Output : 4

Q6.

```
class Writer
{
        public static void write()
        {
                System.out.println("Writing...");
        }
}
class Author extends Writer
{
        public static void write()
        {
                System.out.println("Writing book");
        }
}
public class Programmer extends Author
{
        public static void write()
        {
                System.out.println("Writing code");
        }

        public static void main(String[] args)
        {
                Author a = new Programmer();
                a.write();
        }
}
```

Output : Writing book

```
Q7.class FlipRobo
{
	public static void main(String args[])
	{
		String s1 = new String("FlipRobo");
		String s2 = new String("FlipRobo");
		if (s1 == s2)
			System.out.println("Equal");
		else
			System.out.println("Not equal");
	}
}
```

Output : Not equal

```
Q8.class FlipRobo
{
	public static void main(String args[])
	{
		try
		{
			System.out.println("First statement of try block");
			int num=45/3;
			System.out.println(num);
		}
		catch(Exception e)
		{
			System.out.println("FlipRobo caught Exception");
		}
		finally
		{
			System.out.println("finally block");
		}
		System.out.println("Main method");
	}
}
```

Output :
```
First statement of try block
15
finally block
Main method
```

```
Q9.class FlipRobo
{
	// constructor
	FlipRobo()
	{
		System.out.println("constructor called");
	}

	static FlipRobo a = new FlipRobo(); //line 8

	public static void main(String args[])
	{
		FlipRobo b; //line 12
		b = new FlipRobo();
	}
}
```

Output : constructor called
constructor called

```
Q10.class FlipRobo
{
	static int num;
	static String mystr;
	// constructor
	FlipRobo()
	{
		num = 100;
		mystr = "Constructor";
	}
	// First Static block
	static
	{
		System.out.println("Static Block 1");
		num = 68;
		mystr = "Block1";
	}
	// Second static block
	static
	{
		System.out.println("Static Block 2");
		num = 98;
		mystr = "Block2";
	}
	public static void main(String args[])
	{
		FlipRobo a = new FlipRobo();
		System.out.println("Value of num = " + a.num);
		System.out.println("Value of mystr = " + a.mystr);
	}
}
```

Output : Static Block 1
Static Block 2
100 Constructor