graphixs

BSCCS2005: Graded Assignment Questions with Solutions
Week {2}

{Write general instructions here}

1. What is the value of `str2` at the end of execution of the following Java code? [MCQ:2points]

```
public class StringExample {
    public static void main(String[] args) {
        String str1, str2;
        str1 = "welcome to IITM";
        str2 = str1.substring(0, 11) + "java course" ;
    }
}
```

◯ `java course`

◯ `welcome to IITM java course`

√ `welcome to java course`

◯ `Error in code`

Solution: str1.substring(0, 11) gives "welcome to "
+ is used for string concatenation
"welcome to " + "java course" results in "welcome to java course"

2. What is the output of the following Java code ? [MCQ:2points]

```
public class Print {
    public static void main(String[] args) {
        int[] a = {1, 2, 3};
        int[] b = {1, 2, 3};
        System.out.println(a == b);
    }
}
```

◯ `a == b`

√ `false`

◯ `true`

◯ `{1, 2, 3} == {1, 2, 3}`

Solution: It prints false. The "==" operator compares whether a and b are referring to the same memory location.

3. Consider the Java program below. [MCQ:2 points]

```
class FClass{
    public static void main(String[] args) {
        int i1 = 10, i2 = 29;
        double d;
        d = i2 / i1;
        System.out.print(d + " ");
        d = (double)(i2 / i1);
        System.out.print(d + " ");
        d = (double)i2 / i1;
        System.out.print(d);
    }
}
```

What will be the output?

- ◯ `2.9 2.9 2.9`
- ◯ `2 2.9 2.9`
- √ `2.0 2.0 2.9`
- ◯ `2.0 2.0 2.0`

Solution: In statement `d = i2 / i1;`, since `i2` and `i1` are `int`, it is an integer division and assigns `d` with 2. Since `d` is a `double`, it is printed as 2.0.
In statement `d = (double)(i2 / i1);`, since `i2` and `i1` are `int`, it is an integer division and the result get type-casted to `double`. Thus, `d` would be assigned to 2.0.
In statement `d = (double)i2 / i1;`, `i2` which is of `int` type, type-casted to `double`. Thus, the division becomes floating-point division and `d` would be assigned to 2.9.

4. Identify the correct definition(s) of a `boolean` variable named `flag` in Java from the following. [MSQ:2 points]

◯ `boolean flag = 1;`

√ `boolean flag = true;`

◯ `boolean flag = TRUE;`

◯ `boolean flag = "false";`

Solution: In Java a `boolean` variable can be assigned to either `true` or `false`.

5. Consider the Java program below. [MCQ:2 points]

```
class Point{
    private int x;
    private int y;
    public Point(int x, int z) {
        x = x;
        y = z;
    }
    public void printPint() {
        System.out.println("(" + x + ", " + y + ")");
    }
}
class CClass{
    public static void main(String[] args) {
        Point p = new Point(10, 20);
        p.printPint();
   }
}
```

What will be the output/error?

- ◯ `(0, 0)`
- ◯ `(10, 0)`
- √ `(0, 20)`
- ◯ `(<garbage-value>, <garbage-value>)`
- ◯ `Compiler error: Invalid assignment in constructor`

Solution: In the constructor `public Point(int x, int z)`, for the statements:

```
x = x;
y = z;
```

Here the name of function parameter and data member is same for `x` so inside the constructor `x` refers to the local function parameter and not the object's data member. Such is not the case with the other parameter `z` and hence the initialization for `y` works in usual manner.

6. Consider the Java program below. [MCQ:2 points]

```
import java.util.*;
class Example{
    public static int doSomething(int num) {
        int n = num;
        int total = 0;
        for(int i = 1; i <= n; i++) {
            if (n % i == 0) {
                total = total + i;
            }
        }
        return total;
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int number = sc.nextInt();
        int x = doSomething(number);
   }
}
```

What will `x` represent?

- ◯ Number of multiples of `number`
- ◯ Sum of multiples of `number`
- ◯ Number of factors of `number`
- √ **Sum of factors of** `number`
- ◯ Compiler Error

Solution:

- Program execution begins from `main()` function.
- In main function, user inputs `number`.
- `doSomething()` method is called by passing `number`
- In `doSomething()` value of `number` is assigned to `n`, `i` is initialized to 1 and is incremented in each iteration till `i = n`
- `i` is added to `total`, if `i` divides `n` without any reminder. That means `i` must be factor of `n`.
- Hence, `total` represents sum of factors of `n`.
- `doSomething()` returns `total` and is stored in `x`

7. Consider the Java code given below. [MCQ : 2 points]

```
class  WhileEx2
{
    public static void main(String[] args)
    {
        int i=0;
        while(i>0)
        {
            System.out.println(i);
        }
        do
        {
            System.out.println(i);
        }while(i>0);
    }
}
```

Choose the correct option regarding the given code.

- ◯ Generates no output but program compiles successfully.
- ◯ Generates a compilation error
- ◯ Generates output :
  0
  0
- √ **Generates output : 0**

Solution: In `while` loop, the body of the loop executes when condition is `true`, whereas in `do-while` loop, first time the loop-body always executes once and at the end of the loop condition would be checked.
In above code, `while` loop prints nothing since the condition is failed at the first time, but `do-while` prints 0 as the condition is checked at the end of the loop.

8. Consider the Java code given below. [MCQ : 2 points]

```
class Sample
{
    public static void main (String args[])
    {
        System.out.println(10+20+"IIT Madras");
        System.out.println("IIT Madras"+10+20);
    }
}
```

What will be the output/error?

- ◯ Compilation fails due to `System.out.println(10+20+"IIT Madras");`
- ◯ Compilation fails due to `System.out.println("IIT Madras"+10+20);`
- √ `30IIT Madras`
  `IIT Madras1020`
- ◯ Exception at run time.

Solution: `System.out.println(10+20+"IIT Madras");`
Here both `10` and `20` are integers `10+20` will generate `30`, now `30` integer and a string `IIT Madras` both are concatenated and it generates `30IIT Madras as output`.
`System.out.println("IIT Madras"+10+20)`
Here a string `IIT Madras` is concatenated with an integer `10` and it generates a string `IIT Madras10`, now a string `IIT Madras10` and an integer `20` are concatenated and it generates string `IIT Madras1020` as output.

9. Match the following. [MCQ:2 points]

A. System	I. Prints arguments in a new line.
B out	II. Public class.
C. println()	III. Stream object.
D. braces {...}	IV. Delimits blocks and statement.

◯ A-III, B-II, C-IV, D-I

◯ A-II, B-I, C-III, D-IV

√ **A-II, B-III, C-I, D-IV**

◯ A-I, B-IV, C-II, D-III

Solution:

`System`: A Java program is a collection of classes. System is a public class in Java.
`out`: It is a stream object defined in the System class.
`println()`: Prints arguments in new line like python print()
braces {...} : Delimits blocks and statement. This is similar to the indentation in Python programming.

10. How is it possible to run the main method in Java without creating an object? [MCQ:2 Points]

- ◯ Java is an object oriented programming language. Thus we need to create an object of the main class, and call our `main()` method to run. Hence the question is fallacious.
- √ **The modifier static helps the `main()` method to run independently without creating an object.**
- ◯ The access modifier public helps the `main()` method to run independently without creating an object.
- ◯ The `main()` method is an exception. It is the only method that exist independently without the dynamic creation of an object.

Solution:

The `main()` method in java is declared static. Static methods are the methods in Java that can be called without creating an object of class.

11. Consider the Java code given below. [MCQ:2 points]

```
class FClass
{
    public static void main(String args[])
    {
        int arr[] = {0 , 1, 2, 3, 4, 5, 6, 7, 8, 9};
        int n = 9;
        n = arr[arr[n] / 2];
        System.out.println(arr[n] / 3);
    }
 }
```

Choose the correct option regarding the given code.

- ◯ The program generates output:
  `0`
- √ **The program generates output:**
  `1`
- ◯ The program generates output:
  `1.3333`
- ◯ The program generates a compilation error.

Solution:
We have `int n=9;`
`arr[arr[9]/2] = arr[9/2] = arr[4] = 4`.
Hence, `arr[4]/3 = 4/3 = 1`.

12. Consider the Java code given below. [MCQ:2 points]

(a)
```
public class FClass
{
    public static void main(String args[])
    {
        int x; x = 1;
        if(x) { System.out.println(x); }
        else { System.out.println("2"); }
     }
}
```

Choose the correct option regarding the given code.

- ◯ The program generates output:
  `0`
- ◯ The program generates output:
  `1`
- ◯ The program generates output:
  `2`
- √ **This program will generate an error.**

(b)
```
public class FClass
{
    public static void main(String args[])
    {
        int x=1;
        if(false) { System.out.println(x); }
        else { System.out.println("True"); }
    }
}
```

- √ **The program generates output:**
  `True`
- ◯ The program generates output:
  `False`
- ◯ The program generates output:
  `1`
- ◯ This program will generate an error.

Solution:
(a) `if` is a conditional statement. It accepts `boolean` as a parameter. Here `x` is an integer so this program will yield an error.
(b) If the condition is `false`, then the `else` block gets executed.

13. Consider the Java code given below..

[MCQ:2 points]

```
public class FClass
{
    public static void main(String args[])
    {
        switch(2)
        {
            case 1:
                System.out.println("One");
            case 2:
                System.out.println("Two");
            case 3:
                System.out.println("Three");
            default:
                System.out.println("Default");
                break;
        }
    }
}
```

What will be the output?

- ◯ `One`
- ◯ `Two`
  `Default`
- ◯ `Three`
- √ **None of them**

Solution: The output of the above program is:

```
Two
Three
Default
```

Because in between cases there is no break statement.

14. Match the following. [MCQ:2 points]

A. Multiple constructors	I. constructor with empty arguments
B Default constructor	II. Refers to the current class object
C. Copy constructors	III.Create a new object from an existing one
D. this	IV. Overloading

- √ **A-IV, B-I, C-III, D-II**
- ◯ A-III, B-I, C-I, D-II
- ◯ A-IV, B-II, C-III, D-I
- ◯ A-IV, B-III, C-I, D-II

Solution:
Multiple constructors — Constructor overloading
Default constructor- If no constructor is defined, compiler creates its own. They do no have any arguments
Copy constructor — make a copy of an existing object
this — refers to the current class object

15. Considering the following Java code, choose the correct statement from among the given options regarding this code.

[MCQ:2points]

```
class Employee
{
    int eid;
    String ename;

    public void display()
    {
        System.out.println("eid: "+eid);
        System.out.println("ename: "+ename);
    }
}

class FClass
{
    public static void main(String[] args)
    {
        Employee e1 = new Employee();
        e1.display();
    }
}
```

◯ The code results in a compilation error as constructor is not defined.

◯ Program runs successfully, and the data members `eid` and `ename` contains random garbage value.

√ **Program runs successfully, and the data members `eid` and `ename` contains *0* and *null* respectively.**

◯ None of the above options are correct.

Solution:
If no constructor is defined in a class the default constructor would be executed when an object is created. In JAVA uninitialized variables have values *0* for numeric types and *null* for string types.