INTRODUCTION TO JAVA

LECTURE 4 : CONDITIONALS AND LOOPS

(Error in code)

1. What is the error in this code?

```
byte b = 50;
b = b * 50;
```

a) b can not contain value 2500, limited by its range.

b) * operator has converted b * 50 into int, which can not be converted to byte without casting.

c) b can not contain value 50.

d) No error in this code

SOLUTION DESCRIPTION:

####In 2nd statement, b is multiplied with 50 where b is byte variable and 50 is an integer. So after the multiplication, result comes as an integer value which is 2500. Now we are trying to assign this integer to a byte variable, which will cause the error.

(Find the output)

2.

```
public class  Solution{
    public static void main(String [] args) {
        double a = 6 / 4;
        int b  = 6 / 4;
        double c = a + b;
        System.out.println(c);
    }
}
```

a) 3.0

b) 1.5

c) 2.0

d) 2.5

SOLUTION DESCRIPTION:

####When 6 / 4 is performed, both the operands of / are integer. Hence answer will be an int i.e. 1. When we store it in a (which is double), value of a will be 1.0 and value of b will be 1. Thus a + b will be 2.0.

(Predict the output)

3.

```
public class  Solution{
    public static void main(String [] args) {
        double a = 55.5;
        int b = 55;
        a = a % 10;
        b = b % 10;
        System.out.println(a + " "  + b);
      }
 }
```

a) 5 5

b) 5.5 5

c) 6 5

d) None of these

SOLUTION DESCRIPTION:

####% operator gives remainder. So a % 10 will give us 5.5 and b % 10 will give us 5. Hence output is : 5.5 5

(Predict the output)

4.

```
public class  Solution {
    public static void main(String [] args) {
        int var1 = 5;
        int var2 = 6;
        System.out.print(var1 > var2);
    }
}
```

a) true

b) false

c) 0

d) 1

SOLUTION DESCRIPTION:

####> is a relational operator. So it will give the result as true or false only. var1 is not greater than var2, hence result is false.

(Find output)

5. Find the output of the following code:

```
public static void main(String args[])
{
    int a=10,b=15;
    if(a>b)
    {
        System.out.print("a ");
    }
    else
    {
        System.out.print("b ");
    }
    System.out.print("is greater");
}
```

a) a

b) b

c) a is greater

d) b is greater

SOLUTION DESCRIPTION:

"is greater" is written outside if-else so it would always print

(Find output)

6. Find the output of the following code:

```
int a=50;
```

```
      if(a>10)
      {
          System.out.print("Coding");
      }
      else(a>20)
      {
          System.out.print("Ninjas");
      }
```

a) Coding

b) Ninjas

c) CodingNinjas

d) Compile time error

SOLUTION DESCRIPTION:

else (a>20) is wrong syntax.

We cannot use condition after else.

(Find the output)

7.

```
      public static void main(String args[])
      {
          int x = 5;
          if (x < 6)
              System.out.print("Hello  ");
          if(x == 5){
              System.out.print("Hi  ");
          }
          else{
              System.out.print("Hey ");
          }
      }
```

a) Hello

b) Hi

c) Hello Hi

d) Hello Hey

SOLUTION DESCRIPTION:

As x is equal to 5, it enters in first if condition and prints "Hello". After that, second if condition will be checked and that is true again, so it will next print "Hi". Else part will be skipped.

(Figure out the output)

8.

```
      public static void main(String[] args) {
          int x = 15;
          if(x <= 15){
              System.out.print("Inside if  ");
          }else if(x == 15){
              System.out.print("Inside else if  ");
          }
          System.out.println(x);
      }
```

a) Inside if

b) 15

c) Inside if 15

d) Inside if Inside else if 15

SOLUTION DESCRIPTION:

####First if condition will be checked first, and it evaluates to true. Hence, the statement inside if will be executed, so it will print "Inside if ".

####After that, else if part will be skipped. Because in if-else statements, once a condition is satisfied remaining all the conditions are skipped without evaluation.

####After exit from if-else statement, the last print statement will be executed. Hence it will next print value of x i.e. 15.

####So the output is : "Inside if 15"

(What will be the output)

9.

```
public static void main(String args[]) {
    int var1 = 5;
    int var2 = 6;
    if ((var2 = 1) == var1)
        System.out.print(var2);
    else
        System.out.print(var2 + 1);
}
```

a) 7

b) 1

c) 2

d) 6

SOLUTION DESCRIPTION:

####Inside if condition, we are actually assigning 1 to var2 and then comparing it with var1. So, after assignment, var2 becomes equal to 1 and then we are comparing it with var1 whose value is 5. Both are not equal, hence else part will be executed.

####So the output will be 2, as var2 was updated to value 1 in the if condition.

(Positive Negative)

10. Let a and b are the two integers. Which option can be used to check out of two numbers one is positive and the other is negative

a) a<0&&b>0

b) a>0&&b<0

c) a<0||b<0

d) a*b<0

SOLUTION DESCRIPTION:

Given two integers a and b, the product of two integers is negative means either of a or b is negative.

i.e, If we multiply a*b and the result is less than 0 , it means either a or b is negative.

(Find the output)

11. Find the output of the following code:

```
public static void main (String[] args)
{
    int a=50;
    int b=5;
    if(a/0==b)
    {

        System.out.println("Hello");
    }
    else
    {
        System.out.println("Hi");
    }
}
```

a) Hello

b) Hi

c) Compile time error

d) Run time error

SOLUTION DESCRIPTION:

The syntax of the code is correct but on execution dividing a number by zero gives a run time error.

(Find the output)

12. Find the output of the code :

```
main() {
    int a=50,b=20;
    if(a>b)
    {
        if(a>100)
            print ("Ace");
        if(b<100)
            b=50;
    }
    else if(a==b)
    {
       print ("King");
    }
    else
    {
        print ("Queen");
    }
}
```

a) No output

b) King

c) Queen

d) Ace

SOLUTION DESCRIPTION:

(a>b) is true,So we go inside the if.

(a>100) is false.

(b<100) is true,so it is executed and b is updated to 50.

But no print statement is executed hence no output.

(Find the output)

13. What will be the output of the following code:

```
public static void main (String[] args)
{
   int i=0;
   while(i<10)
   {
       i=i+1;
       System.out.print(i);
       i=i+1;
   }
}
```

ANS: 13579

SOLUTION DESCRIPTION:

The variable i =0. First time the loop executes,it increases the variable by 1 , print is and then again increases it by 1, making it 2. Now, second time, since the condition is true, the variable i would be incaresed to 1 , ie. i=3 and would be printed and so on the loop will continue, till i=9. So the output would be all the odd numbers<10, which is : 13579

(Number of hello)

14.The number of Hello printed on the screen for the following code will be:

```
public static void main (String[] args)
{
    int x=5;
    int y=5;
    while((x=5)==y)
    {
        System.out.println("Hello");
        x++;
        y++;
    }
}
```

a) Infinite

b) Zero

c) One

d) Error

SOLUTION DESCRIPTION:

The loop will run the first time as x=5 and y is also 5. Both are equal so the loop will get executed and after printing "Hello" x and y both will be incremented to 6. Now, if we check the while condition x=5 (x is being assigned the value 5) and y=6 (due to increment). So, x and y are not equal , hence condition is false and the loop will not get executed.

(Number of hello 2)

15. The number of Hello printed on the screen for the following code will be:

```
public static void main (String[] args) {
    int x=5;
    int y=5;
    while(x==y)
    {
        System.out.println("Hello");
        x++;
        y++;
    }
}
```

a) 1

b) Infinite

c) 5

d) 0

SOLUTION DESCRIPTION:

x and y are equal every time . Hence infinite loop.

(Same output)

16. Whichof the following codes gives same output:

```
1) int i=1;
   while(i<5)
   {
       System.out.print(2*i);
       i=i+1;
   }

2) int i=2;
   while(i<10)
   {
       System.out.print(i);
       i=i+2;
   }

3) int i=2;
   while(i<10)
   {
       System.out.print(i);
       i*=2;
   }

4) int i=10;
   while(i>0)
   {
       if(i%2==0)
       {
           System.out.print(10-i);
       }
       i--;
   }
```

a) 2 4

b) 1 2 3 4

c) 1 2

d) 1 2 4

SOLUTION DESCRIPTION:

Output of the codes are:

Code 1: 2468

Code 2: 2468

Code 3: 248

Code 4: 02468

(Primality Check)

17. Can this code be used to check primality of a positive integer:

```
public static void main (String[] args)
{
    Scanner s=new Scanner (System.in);
    int n=s.nextInt();
    boolean isprime=true;
    if(n%2==0)
        isprime=false;
    int i=3;
    while(isprime&&i<n)
    {
        isprime=!(n%i==0);
        i+=2;
    }
    if(isprime)
    {
        System.out.println("Prime");
    }
    else
    {
        System.out.println("Composite");
    }
}
```

a) Yes

b) No

SOLUTION DESCRIPTION:

The code fails for n=2. Otherwise the code runs fine for every other positive integer.

(Find the output)

18. Find the output for the following code:

```
int i=10;
while((i=i-1)>0)
{
    System.out.print(i);
    if(i%5==0)
        return;
}
```

a) 987654321

b) Error

c) 98765

d) 9876

SOLUTION DESCRIPTION:

The code executes for i=9,8,7,6,5 and prints 98765.

At i=5 the if statement is executed and the code terminates.

(Check the error)

19. Will following code generate error ?

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        if(a > 5) {
            int b = 10;
        }
        System.out.println(b);
    }
}
```

a) Yes b) No

SOLUTION DESCRIPTION:

Since int b is declared within the if block, it will not be accessible beyond it. We can see that the statement to print b is written outside the if block, hence it would generate error.

(Check the error)

20. Will following code generate error ?

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        if(a > 5) {
            a = 100;
        }
        System.out.println(a);
    }
}
```

a) Yes **b) No**

SOLUTION DESCRIPTION:

The variable a is accessible within the complete main (). So, the code will not generate any error.

(Check the error)

21. Will following code generate error ?

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        if(a > 5) {
            int a = 100;
        }
        System.out.println(a);
    }
}
```

a) Yes b) No

SOLUTION DESCRIPTION:

Java does not allow us to declare a same variable twice. We can see that int a was declared in the main() and then again it was declared inside the if block, which is not permitted. So, the code will throw the error.

(Check the error)

22. Will following code generate error ?

```
public class Main {
    public static void main(String[] args) {
    for(int i = 0; i < 3; i++) {
        System.out.print(i + " ");
        }
    System.out.print(i + " ");
    }
}
```

a) Yes

b) No

(What is the output)

23. What is the output ?

```
public class Main {
    public static void main(String[] args) {
        int a = 10;
        while(a > 5) {
            int b = 1;
            System.out.print(b + " ");
            a--;
        }
    }
}
```

a) 10 9 8 7 6

b) 1 1 1 1 1

c) Error

d) Infinite loop

SOLUTION DESCRIPTION:

The while loop will be executed 5 times, starting from when a=10 till it become a=6/. Each time the variable b since it is inside the while, would be initialized by 1, so the output will be 1 1 1 1 1

****