Object Oriented Programming with JAVA

Test 2

Name : Sandesh Subedi 'A'

Intake : BSc. IT (Semester III)

Intake Code : NPI2F1909IT

Student ID : NPI000040

Course Name : Object Oriented Development with Java

Course Code : CT038-3-2

Examination Type : Test

Date : 28th July, 2021

Q no. 1)

Ans.

(A) - FileInputStream

Q no. 2)

Ans.

(C) - FileInputStream

Q no. 3)

Ans.

(A) - Constructor of X then Y, finally of Z

Q no. 4)

Ans.

(A) - abstract class, interface

Q no. 5)

Ans.

(D) - Instance variables are initialized to their default values

Q no. 6)

Ans.

(B) - FileNotFoundException

Q no. 7)

Ans.

(A) - brown

Q no. 8)

Ans.

(B) - 30

Q no. 9)

Ans.

(C) - FlowLayout

Q no. 10)

Ans.

(B) - catches

Q no. 11)

Ans.

The three types of exceptions in Java are :

i. **Checked Exceptions / Compile Time Exceptions**

Checked exceptions are the type of exceptions that are checked by the compiler during the time of program compilation. A checked exception includes the exceptions which can be subsisted by Java application if handled appropriately. However, if the program is not well-written, the compiler will display a certain compilation error.

Examples : SQLException, IOException, etc.

ii. Unchecked Exception / Runtime Exception

Unchecked exceptions, also known as 'Runtime Exceptions' are those exceptions which cannot be handled by Java application itself. Unchecked exceptions depends on how precise a programmer write their program. Since this is not controlled or examined by the compiler, faults or inaccuracies from programmer is what results into runtime exceptions.

Examples : ArrayIndexOutOfBoundsException, Arithmetic Exception, etc.

iii. Errors

Errors can be described as exterior issues that are unrecoverable, even with exception handling techniques. Errors are found during runtime which means they are not handled by the compiler. Errors usually occurs when there is any sort of failure in JVM (Java Virtual Machine).

Examples : java.lang.OutOfMemoryError, java.lang.VirtualMachineError, etc.

Q no. 12)

Ans.

```
import java.lang.String;
public class Test2Q12
{
    private int x;
    private String b;
    public Test2Q12(int y, String z)
    {
        x = y;
        b = z;
    }
```

```
public void setX(int y)
{ x = y; }
public void setY(String z)
{ b = z; }
public int getX()
{return x; }
public String getB()
{return b;}

public boolean equals(Object obj1)
{
    if (obj1 instanceof Test2Q12)
    {
        Test2Q12 obj2 = (Test2Q12) obj1;
        if( obj2.getX() == this.getX() && obj2.getB() == this.getB() )
        {
            return true;
        }
        else
        {
            return false;
        }
    }
    else
    {
    System.out.println("Assure both the objects are of same type. i.e, Test2Q12");
    return false;
    }
    }
```

```
    public static void main(String [] args)
    {
        Test2Q12 ob1 = new Test2Q12(10, “Sandesh”);
        Test2Q12 ob2 = new Test2Q12(11, “Sandesh”);
        Test2Q12 ob3 = new Test2Q12(10, “Sandesh”);

        System.out.println(“The object ob1 is equals to ob2 :” + ob1.equals(ob2));
        System.out.println(“The object ob1 is equals to ob3 :” + ob1.equals(ob3));
    }
}
```