SRN □□□□□□□□□□□□

PES University, Bangalore

(Established under Karnataka Act No. 16 of 2013)

UE19CS208A

NOV 2020: INTERNAL ASSESSMENT 2 (Special Topic) B.TECH.

UE19CS208A- Object Oriented Programming with Java

Time: 60min | Answer all questions in the same order | Max Marks: 20

1	a)	Is multiple inheritance allowed in java? If yes illustrate, if not what is the alternative way to achieve the nature of multiple inheritance, illustrate?	5
	Solution	Not allowed, it can be achieved with help of interface.	

```
interface FirstInterface {
  public void myMethod(); // interface method
}

interface SecondInterface {
  public void myOtherMethod(); // interface method
}

class DemoClass implements FirstInterface, SecondInterface {
  public void myMethod() {
    System.out.println("Some text..");
  }
  public void myOtherMethod() {
    System.out.println("Some other text...");
  }
}
```

```
class MyMainClass {

  public static void main(String[] args) {

    DemoClass myObj = new DemoClass();

    myObj.myMethod();

    myObj.myOtherMethod();

  }

}
```

b) Create a user defined package and provide the steps to compile and execute the same. Also provide the way to use the created package in another file. 5

Solution

MyPackageClass.java

```
package mypack;

class MyPackageClass {

  public static void main(String[] args) {

    System.out.println("This is my package!");

  }

}
```

Save the file as MyPackageClass.java, and compile it:

```
C:\Users\Your Name>javac MyPackageClass.java
```

Then compile the package:

```
C:\Users\Your Name>javac -d . MyPackageClass.java
```

This forces the compiler to create the "mypack" package.

The `-d` keyword specifies the destination for where to save the class file. You can use any directory name, like c:/user (windows), or, if you want to keep the package within the same directory, you can use the dot sign "`.`", like in the example above.

When we compiled the package in the example above, a new folder was created, called "mypack".

To run the MyPackageClass.java file, write the following:

```
C:\Users\Your Name>java mypack.MyPackageClass
```

The output will be:

```
This is my package!
```

Example for built in packages:java,lang etc.

2	a)	Write a java program to copy the contents of one file to another.	5

Solution

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class CopyExample
{
    public static void main(String[] args)
    {
        FileInputStream instream = null;
        FileOutputStream outstream = null;

        try{
            File infile =new File("C:\\MyInputFile.txt");
            File outfile =new File("C:\\MyOutputFile.txt");

            instream = new FileInputStream(infile);
            outstream = new FileOutputStream(outfile);

            byte[] buffer = new byte[1024];

            int length;
            /*copying the contents from input stream to
             * output stream using read and write methods
             */
            while ((length = instream.read(buffer)) > 0){
                outstream.write(buffer, 0, length);
            }

            //Closing the input/output file streams
            instream.close();
            outstream.close();

            System.out.println("File copied successfully!!");

        }catch(IOException ioe){
                ioe.printStackTrace();
         }
    }
}
```

b) Write a java program to demonstrate non static inner class. 5

Solution

```
public class Example1
{
    public static void main(String[] args)
    {
        Domain d1 = new Domain("RainBow");
        Domain.member m1 = d1.new member("john");
        Domain.member m2 = d1.new member("peter");
        Domain d2 = new Domain("Sea");
        Domain.member m3 = d2.new member("mary");
        m1.disp();
        m2.disp();
        m3.disp();
    }
}
class Domain
{
    private String name;
    public Domain(String name) { this.name = name; }
    class member
    {
        private String memberName;
        public member(String memberName)
        {
            this.memberName = memberName;
        }
        public void disp()
        {
            System.out.println("name : " + name + " member : " + memberName);

        }
    }
}
```