Name: Mohd Hamza Abbasi

PRN: 22070122125

Batch: CSE B

Question:

1. WAP to demonstrate static block.

```
public class Bank
{
     static int nooftransaction=0;
     public void withdraw(double amount)
     {
          nooftransaction++;
          System.out.println(amount+" Rs. debited via
atm.");
     }
     public void withdraw(double amount, String upiid)
     {
          nooftransaction++;
          System.out.println(amount+" Rs. debited via
UPI : "+upiid);
     }
     public void withdraw(double amount, long
accountno)
     {
          nooftransaction++;
          System.out.println(amount+" Rs. debited via
accno. : "+accountno);
     }
     public void deposit(double amount)
     {
          nooftransaction++;
          System.out.println(amount+" Rs. credited via
atm.");
     }
     public void deposit(double amount, String upiid)
     {
          nooftransaction++;
```

```
        System.out.println(amount+" Rs. credited via
UPI : "+upiid);
    }
    public void deposit(double amount, long
accountno)
    {
        nooftransaction++;
        System.out.println(amount+" Rs. credited via
accno. : "+accountno);
    }
    public static void total()
    {
        System.out.println("Total number of
transaction : "+nooftransaction);
    }
    public static void main(String[]args)
    {
        Bank upi=new Bank();
        Bank atm=new Bank();
        Bank netbanking=new Bank();
        upi.withdraw(540,"7905011947@ibl");
        upi.deposit(720,"9336865880@pz");
        atm.withdraw(500);
        atm.deposit(700);
        netbanking.withdraw(1000,723862323);
        netbanking.deposit(500,983274283);
        total();
    }
}
```

```
540.0 Rs. debited via UPI : 7905011947@ibl
720.0 Rs. credited via UPI : 9336865880@pz
500.0 Rs. debited via atm.
700.0 Rs. credited via atm.
1000.0 Rs. debited via accno. : 723862323
500.0 Rs. credited via accno. : 983274283
Total number of transaction : 6
```

2. WAP to print the number of objects created for a class.

```
public class ObjectCounter {
    static int count = 0;
    public ObjectCounter()
    {
        count++;
    }
    public static void main(String[] args)
    {
        ObjectCounter obj1 = new ObjectCounter();
        ObjectCounter obj2 = new ObjectCounter();
        ObjectCounter obj3 = new ObjectCounter();
        System.out.println("Number of objects
created: " + count);
    }
}
```

```
Number of objects created: 3
```

3. In Box class, add member function to find values of private variables.

```
4.public class Box
5.{
6.    private double length;
7.    private double width;
8.    private double height;
9.    public void setDimensions(double l, double w,
  double h)
10.       {
11.           length = l;
12.           width = w;
13.           height = h;
14.       }
15.       public double getLength()
16.       {
17.           return length;
18.       }
19.       public double getWidth()
20.       {
21.           return width;
22.       }
23.       public double getHeight()
24.       {
```

```
25.             return height;
26.         }
27.         public double calculateVolume()
28.         {
29.             return length * width * height;
30.         }
31.         public static void main(String[] args)
32.         {
33.             Box myBox = new Box();
34.             System.out.println("Length :
  "+myBox.getLength());
35.             System.out.println("Width :
  "+myBox.getWidth());
36.             System.out.println("Height :
  "+myBox.getHeight());
37.             myBox.setDimensions(5, 4, 3);
38.             System.out.println("Length :
  "+myBox.getLength());
39.             System.out.println("Width :
  "+myBox.getWidth());
40.             System.out.println("Height :
  "+myBox.getHeight());
41.         }
42.     }
```

```
Length : 0.0
Width : 0.0
Height : 0.0
Length : 5.0
Width : 4.0
Height : 3.0
```

4. In Box class, add member function to print volume of box.

```
    public class Box
    {
        private double length;
        private double width;
        private double height;
        public void setDimensions(double l, double w,
double h)
        {
```

```
            length = l;
            width = w;
            height = h;
        }
        public double getLength()
        {
            return length;
        }
        public double getWidth()
        {
            return width;
        }
        public double getHeight()
        {
            return height;
        }
        public double calculateVolume()
        {
            return length * width * height;
        }
        public static void main(String[] args)
        {
            Box myBox = new Box();
            myBox.setDimensions(5, 4, 3);
            System.out.println("Volume of the box is:
" + myBox.calculateVolume());
        }
    }
```

```
Volume of the box is: 60.0
```

5. Execute Inner class-related program conducted during Saturday session

```
class Operand
{
    private int op1;
    private int op2;
    public int getop1()
    {
        return op1;
    }
```

```
    public void setop1(int op1)
    {
        this.op1=op1;
    }
    public int getop2()
    {
        return op2;
    }
    public void setop2(int op2)
    {
        this.op2=op2;
    }
    public void add()
    {
        System.out.println("Addition =
"+(this.op1+this.op2));
    }
    public void sub()
    {
        System.out.println("Difference = "+(this.op1-
this.op2));
    }
    public void mul()
    {
        System.out.println("Product =
"+(this.op1*this.op2));
    }
    public void div()
    {
        System.out.println("Questiont =
"+(this.op1/this.op2));
    }
}
public class Calculate
{
    public static void main(String[]args)
    {
        Operand ob=new Operand();
        ob.setop1(30);
        ob.setop2(10);
        ob.add();
        ob.sub();
        ob.mul();
        ob.div();
```

```
    }
}
```

```
Addition = 40
Difference = 20
Product = 300
Questiont = 3
```

6. Study questions given in below link:

https://www.javainuse.com/misc/inner-interview-questions

Take a screenshot, paste in word file for submission

Q: What is Inner Class in Java?

A: In Java similar methods and variables of a class, we can have a class as member of another class . Declaring a class within another is allowed in Java. The class written within is called the nested class, and the class that holds the inner class is called the outer class. We use inner classes to logically group classes and interfaces in one place so that it can be more readable and maintainable.

```
class Java_Outer_Demo {
   class Java_Inner_Demo {
   }
}
```

Q: What are the advantages of using Inner Class in Java?

A: The Advantages of using Polymer are as follows-

- Nested classes represent a special type of relationship that is it can access all the members (data members and methods) of outer class including private.
- Nested classes are used to develop more readable and maintainable code because it logically group classes and interfaces in one place only.
- Code Optimization: It requires less code to write.

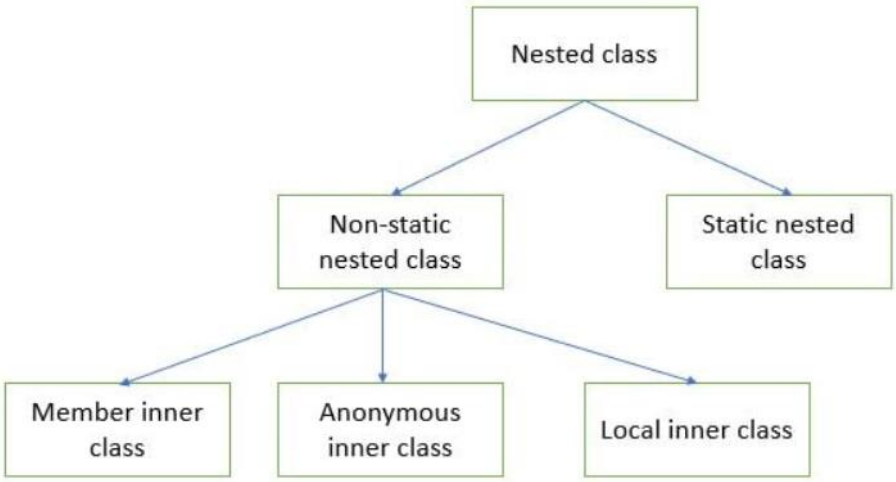
Q: What are the types of nested classes?

A: Nested classes can be divided into two types -

- Non-static nested classes - These are the non-static members of a class.
- Static nested classes - These are the static members of a class.

Non-static class can be further classified into 3 types

- Member inner class
- Anonymous inner class
- Local inner class



Q: What is anonymous inner class in Java?

A: It is an inner class without a name and for which only a single object is created. An anonymous inner class can be useful when making an instance of an object with certain "extras"" such as overloading methods of a class or interface, without having to actually subclass a class. Anonymous inner classes are useful in writing implementation classes for listener interfaces in graphics programming. Anonymous inner class can be created in 2 ways-

- Class (may be abstract or concrete)
- Interface

```
button.addActionListener(new ActionListener() {
      public void actionPerformed(ActionEvent e) {
        JFileChooser fileChooser = new JFileChooser();
        int returnVal = fileChooser.showOpenDialog(null);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
          System.out.println(fileChooser.getSelectedFile().getName());
        }
      }
    });
```

Q: Why can outer Java classes access inner class private members?

A: The inner class is just a way to cleanly separate some functionality that really belongs to the original outer class. They are intended to be used when you have 2 requirements: Some piece of functionality in your outer class would be most clear if it was implemented in a separate class. Even though it's in a separate class, the functionality is very closely tied to way that the outer class works. Given these requirements, inner classes have full access to their outer class. Since they're basically a member of the outer class, it makes sense that they have access to methods and attributes of the outer class -- including privates.

7. WAP to enter an integer value, a character value, a double value and store them in primitive data types. Use Wrapper class to convert these primitive data types to objects of the class.

```
import java.util.Scanner;
public class PrimitiveToWrapper {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter an integer value: ");
        int intValue = scanner.nextInt();
        Integer integerValue =
Integer.valueOf(intValue);

        System.out.print("Enter a character value:
");
        char charValue = scanner.next().charAt(0);
        Character characterValue =
Character.valueOf(charValue);

        System.out.print("Enter a double value: ");
        double doubleValue = scanner.nextDouble();
        Double doubleObject =
Double.valueOf(doubleValue);

        System.out.println("Primitive Integer Value:
" + intValue);
        System.out.println("Integer Object: " +
integerValue);

        System.out.println("Primitive Character
Value: " + charValue);
        System.out.println("Character Object: " +
characterValue);

        System.out.println("Primitive Double Value: "
+ doubleValue);
        System.out.println("Double Object: " +
doubleObject);

        scanner.close();
    }
}
```

```
Enter an integer value: 56
Enter a character value: g
Enter a double value: 68.666
Primitive Integer Value: 56
Integer Object: 56
Primitive Character Value: g
Character Object: g
Primitive Double Value: 68.666
Double Object: 68.666
```

8. WAP to print ASCII values

```java
import java.util.Scanner;
public class ASCIIPrinter
{
    public static void main(String[] args)
    {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a character: ");
        char ch = scanner.next().charAt(0);
        int asciiValue = (int) ch;
        System.out.println("ASCII value of " + ch + "
is " + asciiValue);
        scanner.close();
    }
}
```

```
Enter a character: a
ASCII value of a is 97
```