Tanishq Chauhan
21C3184
CS-B

Chauhan

Object Oriented Programming (CER3C2)
Mid Semester Test - III

* Answer of Q. No. 4

(a) • Java finally block is a block used to execute important code such as closing the connection etc.
Java finally block is always executed whether an exception is handled or not. Therefore, it contains all the necessary statements that need to be printed regardless of the exception occurs or not.
The finally block is used to disply information or to execute statements, which needs to execute irrespective of the catch block.
The purpose of finally block is to execute code of finally block when try blocks exit. This ensures that the finally block is executed even if an unexpected exception occurs.
But finally is useful for more than just exception handling. It allows the programmer to avoid having cleanup code.

Scenario in which the code in finally block will not executed:-

If the JVM exits while the try or catch code is being executed, then the finally block may not execute.
If the thread executing the try or catch code is interreputed or killed, the finally block may not execute even though the application as whole continoes.
When the system.exit() method is called in try block

before the execution of finally block, finally block will not be executed.

(b) Advantages of Synchronized block over Synchronized Method :-

Synchronized block reduce scope of lock but synchronized method's scope of lock is whole method.

Synchronized block has better performance as only critical section is locked but synchronized method has poor performance than block.

Synchronized block can throw NULL pointer exception but synchronized method doesn't throw.

Synchronized block provide granular control over lock but synchronized method lock either on current object represented by their or class level lock.

Synchronized Block : synchronized (this) { }
Synchronized Method : public synchronized void fun() { }

Tanishq Chauhan
21C3184
CS-B

Chauhan

* Answer of Q. No. 1

```
import java.util.scanner;

class Temperature extends Exception
{
    public Temperature (String str)
    {
        super (str);
    }

  public class TestCustomException
  {
      static void check (float temperature)
         throws Temperature
      {
        if(temperature <94.00)
        {
           throw new Temperature ("Temperature Below
                                          Normal");
        }

        else if
          ((temperature > = 94.00 && temperature
                               < = 100.00)
          {
             throw new Temperature ("Normal Temperature");

          }
```

Tanishq Chauhan
21C3184
CS-B

Chauhan

```
else
     { throw new Temperature ("High Temperature");

     }
    }

public static void main (String args [])
{
   system.out.println ("Enter a temperature : ");
   Scanner sc = new Scanner (System.in);
   Float temp;
   temp = sc.nextFloat();

   try
        { check (temp);

        }

   catch (Temperature t)

   {
        System.out.println ("Exception caught :" +t);

   }
  }
 }
```

Tanishq Chauhan
21C3184
CS-B

* Answer of Q. No. 2

```
import java.util.scanner;
class Even extends Thread {
                    public int n1,n2;
                    public Even ( int n1, int n2)

                    {
                        this.n1 =n1
                        this.n2 =n2;
                    }

                   public void run()

                   {
                     try
                       {
                         if (n1%2!=0){
                                       n1++;
                                     }
                        for(int i=n1; i<=n2; i=i+2)
                        {
                          System.out.println("Even thread:"
                                                          +i);
                          sleep(500);
                        }
                     }
                     catch(Interrupted Exception e)
                     {
                      system.out.print("Even thread was
                                    interrupted");
                     }}}
```

Tanishq Chauhan
21C3184
CS-B

Chauhan

```
class Odd extents Thread
{
    public int n1, n2;
    public odd (int n1, int n2)
     {
         this.n1 = n1;
         this.n2 = n2;
     }

  public void run()
   {
      try
         { if (n1%2==0)
                            {
                                 n1++;
                            }

                 for (int i=n1; i<n2; i=i+2)
                 {
                   system.out.println ("Odd thread:"+i);
                   sleep (500);
                 }
               }
               catch (Interrupted Exception e)
             {
               system.out.println(" Odd thread was intropped");

             }
          } }
```

Tanishq Chauhan
2103184
CS-B

Chauhan

```
class display
{
   public static void main (string args [])
   {
      Scanner input = new Scanner (system.in);
        int n1 = input.next Int();
        int n2 = input.next Int();
         new Even (n1, n2). start();
         new Odd (n1,n2. start();
      }
  }
```