

5. Write a java program to illustrate the following exceptions

- i. Arithmetic Exception.
- ii. String out of Bound Exception.

// Java program to demonstrate ArithmeticException

class ArithmeticException\_Demo

```
{  
    public static void main(String args[])  
    {  
        try {  
            int a = 10, b = 0;  
            int c = a/b; // cannot divide by zero  
            System.out.println ("Result = " + c);  
        }  
        catch(ArithmeticException e) {  
            System.out.println ("Can't divide a number by 0");  
        }  
    }  
}
```

// Java program to demonstrate StringIndexOutOfBoundsException

class StringIndexOutOfBounds\_Demo

```
{  
    public static void main(String args[])  
    {  
        try {  
            String a = "R V College of Engineering ";  
            char c = a.charAt(28); // accessing 25th element  
            System.out.println(c);  
        }  
    }  
}
```

```

        catch(StringIndexOutOfBoundsException e) {
            System.out.println("StringIndexOutOfBoundsException");
        }
    }
}

```

6. Write a Java program in to declare an interface called Bank, where SBI, and PNB inherits these interfaces through class TestInterface2. Calculate the rate of interest of each bank and print the result.

```

interface Bank{
    float rateOfInterest();
}

class SBI implements Bank{
    public float rate Of Interest(){return 9.15f;}
}

class PNB implements Bank{
    public float rate Of Interest(){return 9.7f;}
}

class TestInterface2{
    public static void main(String[] args){
        Bank b=new SBI();
        System.out.println("ROI: "+b.rateOfInterest());
        Bank b1 =new PNB ();
        System.out.println("ROI: "+b1.rateOfInterest());
    }
}

```

7. Write a program to make a package balance which has account class with display\_balance method in it. Import balance package in another program to access display\_balance method of account class.

**Account.java**

```
package Balance;

import java.util.Scanner;

public class Account {
    int curBalance, amt;

    public Account() {
        curBalance = 500;
    }

    void deposit() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the amount :");
        amt = s.nextInt();
        curBalance += amt;
        System.out.println("Current balance is : " + curBalance);
    }

    void withdraw() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the amount :");
        amt = s.nextInt();

        try {
            if ((curBalance - amt) < 500)
                throw new LessBalanceException(amt);
            curBalance -= amt;
            System.out.println("\nBalance left : " + curBalance);
        } catch (LessBalanceException e) {
            System.out.println(e);
        }
    }
}
```

```
void display_balance() {  
    System.out.println("Balance in your a/c :"+ curBalance);  
}  
}
```

```
class LessBalanceException extends Exception {  
    int amt;  
    LessBalanceException(int x) {  
        System.out.println("Balance is less :"+ amt);  
    }  
}
```

MainProgram.java

```
package Balance;  
  
import java.util.Scanner;  
  
public class MainProgram {  
    public static void main(String[] args) {  
        int ch;  
  
        Scanner s = new Scanner(System.in);  
  
        Account a = new Account();  
  
        while (true) {  
            System.out.println("1:Deposit\t2:Withdraw\t3:Balance\t4:Exit\n");  
            System.out.println("Enter your choice:");  
  
            ch = s.nextInt();  
  
            switch (ch) {  
                case 1:  
                    a.deposit();  
                    break;  
                case 2:  
                    a.withdraw();  
                    break;  
                case 3:  
                    a.display_balance();
```

```

break;
case 4:
return;
default:
System.out.println("Invalid choice\n");
return;
}
}
}
}

```

7. Write a JAVA program to create five threads with different priorities. Send two threads of the highest priority to sleep state. Check the aliveness of the threads and mark which is long lasting.

class ThreadClass implements Runnable

```

{
long click=0;
Thread t;
private volatile boolean running =true;
public ThreadClass(int p)
{
t=new Thread(this);
t.setPriority(p);
}
public void run()
{
while(running)
{
click++;

```

```
}  
}  
public void stop()  
{  
    running =false;  
}  
public void start()  
{  
    t.start();  
}  
}
```

```
public class Demo {  
    public static void main(String args[])  
    {  
        Thread.currentThread().setPriority(Thread.MAX_PRIORITY);  
        ThreadClass hi1=new ThreadClass(Thread.NORM_PRIORITY + 2);  
        ThreadClass hi2=new ThreadClass(Thread.NORM_PRIORITY -2);  
        ThreadClass hi3=new ThreadClass(Thread.NORM_PRIORITY + 3);  
        ThreadClass hi4=new ThreadClass(Thread.NORM_PRIORITY - 3);  
        ThreadClass hi5=new ThreadClass(Thread.NORM_PRIORITY +4);  
        hi1.start();  
        hi2.start();  
        hi3.start();  
        hi4.start();  
        hi5.start();  
        System.out.println("thread one is alive:" +hi1.t.isAlive());  
        System.out.println("thread two is alive:" +hi2.t.isAlive());  
        System.out.println("thread three is alive:" +hi3.t.isAlive());  
        System.out.println("thread four is alive:" +hi4.t.isAlive());  
        System.out.println("thread four is alive:" +hi5.t.isAlive());  
    }  
}
```

```
try
{
    hi5.t.sleep(1000);
    hi3.t.sleep(1000);
}
catch(InterruptedException e){
    System.out.println("main thread interrupted");
}

hi1.stop();
hi2.stop();
hi3.stop();
hi4.stop();
hi5.stop();

try
{
    System.out.println("waiting for threads to finish");
    hi1.t.join();
    hi2.t.join();
    hi3.t.join();
    hi4.t.join();
    hi5.t.join();
}
catch(InterruptedException e)
{
    System.out.println("main thread interrupted");
}

System.out.println("priority of thread1:" +hi1.t.getPriority());
System.out.println("priority of thread2:" +hi2.t.getPriority());
System.out.println("priority of thread3:" +hi3.t.getPriority());
System.out.println("priority of thread4:" +hi4.t.getPriority());
System.out.println("priority of thread5:" +hi5.t.getPriority());
System.out.println("thread one is alive:" +hi1.t.isAlive());
```

```
System.out.println("thread two is alive:" +hi2.t.isAlive());  
System.out.println("thread three is alive:" +hi3.t.isAlive());  
System.out.println("thread four is alive:" +hi4.t.isAlive());  
System.out.println("thread five is alive:" +hi5.t.isAlive());  
System.out.println("main thread exiting");  
}  
}
```

8.