1. Write a program in Java to develop user defined exception for 'Divide by Zero'error.

Solution:

```
public class A {
   public static void main(String[] args)
   try{
   int a=10;
   int b=0;
   int c=a/b;
      System.out.println(""+c);
   }
   catch(ArithmeticException e){
      System.out.println(e);
 }
      System.out.println("Last line of the program");
}
}
```

Output:

```
java.lang.ArithmeticException: / by zero
Last line of the program
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Write a program in Java to demonstrate multiple try block and multiple catch exception.

Solution:

```
public class A {
   public static void main(String[] args) {
   try{
   int a=10;
   int b=0;
   int c=a/b;
      System.out.println(""+c);
   }
   catch(ArithmeticException e){
      System.out.println(e);
 }
   try{
   int a[]=new int[2];
   a[0]=5;
   a[1]=10;
   a[2]=20;
      System.out.println(""+a[2]);
   }
   catch(ArrayIndexOutOfBoundsException ie){
      System.out.println(ie);
 }
      System.out.println("Last line of the program");
}
}
```

Output:

```
java.lang.ArithmeticException: / by zero
java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for length 2
Last line of the program
BUILD SUCCESSFUL (total time: 0 seconds)
```

3. Write a java program using nested try-catch blocks.

Solution:

```
public class A {
public static void main(String[] args) {
      try{
       try{
         int a=5;
         int b=0;
         int c=a/b;
        System.out.println(c);
       }
       catch(ArithmeticException e1){
           System.out.println(e1);
       }
       try{
          int a[]=new int[2];
          a[0]=5;
          a[1]=8;
          a[2]=9;
```

```
            System.out.println(a[2]);
        }
        catch(ArrayIndexOutOfBoundsException e2){
            System.out.println(e2);
        }
    }
    catch(Exception e3){
            System.out.println(e3);
    }
}
```

Output:

```
java.lang.ArithmeticException: / by zero
java.lang.ArrayIndexOutOfBoundsException: Index 2 out of bounds for length 2
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Write a program that executes two threads. One thread displays "Thread1" every 2,000 milliseconds, and the other displays "Thread2" every 4,000 milliseconds. Create the threads by extending the Thread class.

Solution:

```
public class Thread1 extends Thread {
```

```
    public void run(){
        for(int i=0;i<5;i++){
            System.out.println(""+i);
            try{
                sleep(2000);
            }catch(InterruptedException e){
                e.printStackTrace();
            }
        }
    }
}

public class Thread2 extends Thread {
    public void run(){
        for(int i=0;i<5;i++){
            System.out.println(""+i);
            try{
                sleep(4000);
            }catch(InterruptedException e1){
                e1.printStackTrace();
            }
        }
    }
}

public class Test {
    public static void main(String[] args) {
```

```
        Thread1 t1=new Thread1();
        Thread2 t2=new Thread2();
        t1.start();
        t2.start();
    }
}
```

Output:

```
run:
0
0
1
2
1
3
2
4
3
4
BUILD SUCCESSFUL (total time: 20 seconds)
```

5. Write a program in Java to demonstrate use of synchronization of threads when multiple threads are trying to update common variable.

Solution:

```
public class Table {
synchronized void printTable(int n)   {
for(int i = 1; i <= 5; i++){
 System.out.println(""+n * i);
 try{
 Thread.sleep(500);
```

```
 }
catch(InterruptedException ie){
 System.out.println(ie);
 }
 }
}
}

public class Thread1 extends Thread {
Table t;
Thread1(Table t){
 this.t = t;
}
public void run(){
 t.printTable(2);
 }
}

public class Thread2 extends Thread {
 Table t;
Thread2(Table t){
 this.t = t;
}
public void run(){
 t.printTable(10);
 }
}
```

```
public class Test {
public static void main(String[] args) {
Table s = new Table();
Thread1 t1=new Thread1(s);
Thread2 t2=new Thread2(s);
t1.start();
t2.start();
}
}
```

Output:

```
2
4
6
8
10
10
20
30
40
50
BUILD SUCCESSFUL (total time: 5 seconds)
```

6.Wap to print elements of an array in reverse order.

Solution:

```
import java.util.*;
public class RevArray {
```

```
    public static void main(String[] args) {
        int i,n;
        int a[];
        Scanner sc=new Scanner(System.in);
         System.out.println("enter the size of an arry::");
         n=sc.nextInt();
         a=new int[n];
         for(i=0;i<n;i++){
             System.out.println("enter a["+i+"]:");
             a[i]=sc.nextInt();
         }
         System.out.println("Reverse Array: ");
         for(i=n-1;i>=0;i--){
             System.out.println(""+a[i]);
         }
    }
}
```

Output:

```
enter the size of an arry::
5
enter a[0]:
1
enter a[1]:
2
enter a[2]:
3
enter a[3]:
4
enter a[4]:
5
Reverse Array
5
4
3
2
1
BUILD SUCCESSFUL (total time: 14 seconds)
```

7. WAP to read values in two-dimensional array and print them in matrix form.

Solution:

```
import java.util.*;
public class ARRay {
   public static void main(String[] args) {
      Scanner sc =new Scanner(System.in);
      int i,n,j;
      int size;
      System.out.println("Enter the size:");
      size=sc.nextInt();
```

```
int a[][]=new int[size][size];
        System.out.println("Enter the Element:");
        for(i=0;i<a.length;i++){
            for(j=0;j<a.length;j++){
            a[i][j]=sc.nextInt();
        }
        }
        for(i=0;i<a.length;i++){
            for(j=0;j<a.length;j++){
            System.out.println("enter a["+i+"]["+j+"]:"+a[i][j]+"\t");
        }
        }
        System.out.println("");
    }
}
```

Output:

```
Enter the size:
2
Enter the Element:
4
5
6
9
enter a[0][0]:4
enter a[0][1]:5
enter a[1][0]:6
enter a[1][1]:9

BUILD SUCCESSFUL (total time: 10 seconds)
```

8. WAP to store numbers in 4 X 4 matrix in a two-dimensional array. Find the sum of the numbers of each row and the sum of numbers of each column of the matrix.

Solution:

```
import java.util.*;
public class ArrayDemo1{
   public static void main(String[] args) {
      Scanner sc =new Scanner(System.in);
      int i,n,j;
      int size;
      System.out.println("Enter the size:");
      size=sc.nextInt();
      int a[][]=new int[size][size];
      System.out.println("enter the element:");
      for(i=0;i<a.length;i++){
         for(j=0;j<a.length;j++){
         a[i][j]=sc.nextInt();
      }
      }
      for(i=0;i<a.length;i++){
         for(j=0;j<a.length;j++){
       System.out.println("enter a["+i+"]["+j+"]:"+a[i][j]);
      }
      }
      int b[][]=new int[size][size];
         for(i=0;i<b.length;i++){
         for(j=0;j<b.length;j++){
```

```
        b[i][j]=sc.nextInt();
      }
      }
      for(i=0;i<b.length;i++){
        for(j=0;j<b.length;j++){
       System.out.println("enter b["+i+"]["+j+"]:"+b[i][j]);
      }
      }
       int c[][]=new int[size][size];
      for(i=0;i<c.length;i++){
        for(j=0;j<c.length;j++){
       System.out.println("enter c["+i+"]["+j+"]:"+(a[i][j]+b[i][j])+"\t");
      }
        System.out.println();
      }
   }
}
```

Output:

Enter the size:

4

enter the element:

1

2

3

4

5

6

7

8

9

4

5

6

1

2

3

6

enter a[0][0]:1

enter a[0][1]:2

enter a[0][2]:3

enter a[0][3]:4

enter a[1][0]:5

enter a[1][1]:6

enter a[1][2]:7

enter a[1][3]:8

enter a[2][0]:9

enter a[2][1]:4

enter a[2][2]:5

enter a[2][3]:6

enter a[3][0]:1

enter a[3][1]:2

enter a[3][2]:3

enter a[3][3]:6

1

2

3

6

5

4

9

8

7

7

8

9

4

5

6

2

enter b[0][0]:1

enter b[0][1]:2

enter b[0][2]:3

enter b[0][3]:6

enter b[1][0]:5

enter b[1][1]:4

enter b[1][2]:9

enter b[1][3]:8

enter b[2][0]:7

enter b[2][1]:7

enter b[2][2]:8

enter b[2][3]:9

enter b[3][0]:4

enter b[3][1]:5

enter b[3][2]:6

enter b[3][3]:2

enter c[0][0]:2

enter c[0][1]:4

enter c[0][2]:6

enter c[0][3]:10

enter c[1][0]:10

enter c[1][1]:10

enter c[1][2]:16

enter c[1][3]:16

enter c[2][0]:16

enter c[2][1]:11

enter c[2][2]:13

enter c[2][3]:15

enter c[3][0]:5

enter c[3][1]:7

enter c[3][2]:9

enter c[3][3]:8

BUILD SUCCESSFUL (total time: 58 seconds)