Assignment 6

1. Write a program to handle the ArithmeticException.

Solution

```
public class Ques1 {
    public static void main(String[] args) {
        int i = 10;
        try{
            i/=0;
        }catch(RuntimeException e){
            System.out.println(e);
        }
    }
}
```

Output

```
PS F:\suvo\iem 5th sem\oops lab\a 6> cd "f:\suvo\iem 5th sem\oops lab\a 6\" ; if ($?) { javac Ques1.java } ; if ($?) { java Ques1 }
java.lang.ArithmeticException: / by zero
```

2. Write a program for multiple catch to fire ArrayIndexOutOfBoundsException and StringIndexOutOfBoundsException both.

Solution

```
public class Ques2 {
public static void main(String[] args) {
int[] A = {1,2,3,4,5};
try{
System.out.println(A[6]);
} catch(RuntimeException e){
System.out.println(e);
}
String S = "Computer";
try{
System.out.println(S.charAt(10));
} catch(RuntimeException e){
System.out.println(e);
}
}
}
```

Output

```
PS F:\suvo\iem 5th sem\oops lab\a 6> cd "f:\suvo\iem 5th sem\oops lab\a 6\" ; if ($?) { javac Ques2.java } ; if ($?) { java Ques2 }
java.lang.ArrayIndexOutOfBoundsException: 6
java.lang.StringIndexOutOfBoundsException: String index out of range: 10
```

3. Write a program to fire the NegativeArraySize exception.

Solution

```
public class Ques3 {
public static void main(String[] args) {
try{
int[] A = new int[-1];
}catch(RuntimeException e){
        System.out.println(e);
    }
}
}
```

Output

```
PS F:\suvo\iem 5th sem\oops lab\a 6> cd "f:\suvo\iem 5th sem\oops lab\a 6\" ; if ($?) { javac Ques3.java } ; if ($?) { java Ques3 }
java.lang.NegativeArraySizeException
```

4. Define an object reference and initialize it to null. Try to call a method through this reference. Now wrap the in a try-catch clause to catch the exception.

Solution

```
class A{
```

```
private int x;
public A(int x){
this.x=x;
}
public int getX(){
return this.x;
}
}
public class Ques4 {
public static void main(String[] args) {
A alpha = null;
try{
System.out.println(alpha.getX());
}catch(RuntimeException e){
System.out.println(e);
}
}
}
```

Output

```
PS F:\suvo\iem 5th sem\oops lab\a 6> cd "f:\suvo\iem 5th sem\oops lab\a 6\" ; if ($?) { javac Ques4.java } ; if ($?) { java Ques4
java.lang.NullPointerException
```

5. Write a program in Java to create a user defined exception named PayOutOfBoundsException (Provided the monthly salary of a person is less than Rs. 10,000) and fire the exception.

Solution

```
class PayOutOfBoundsException extends RuntimeException{
public PayOutOfBoundsException(String e){ super(e); }
}
public class Ques5 {
public static void main(String[] args) {
int salary = 9000;
try{
if(salary<10000){
throw new PayOutOfBoundsException("Provided monthly salary of a person is less than
Rs. 10,000");
} else {
System.out.println("Monthly Salary is "+salary);
}catch(RuntimeException e){
System.out.println(e);
        }
    }
}
```

Output

```
PS F:\suvo\iem 5th sem\oops lab\a 6> cd "f:\suvo\iem 5th sem\oops lab\a 6\" ; if ($?) { javac Ques5.java } ; if ($?) { java Ques5 }
PayOutOfBoundsException: Provided monthly salary of a person is less than    Rs. 10,000
```

6. Write a program to fire any checked exception manually using 'throw' keyword.

Solution

```
public class Ques6 {
public static void main(String[] args) {
int a=10;
int b=0;
try{
if(b==0){
throw new ArithmeticException("Cannot Divide by Zero");
}else{
System.out.println("The Value: "+a/b);
}
}catch(RuntimeException e){
System.out.println(e);
}
}
```

```
}
```

Output

```
PS F:\suvo\iem 5th sem\oops lab\a 6> cd "f:\suvo\iem 5th sem\oops lab\a 6\" ; if ($?) { javac Ques6.java } ; if ($?) { java Ques6 }
java.lang.ArithmeticException: Cannot Divide by Zero
```

7. Create a class with two methods, f( ) and g( ). In g( ), throw an exception of a new type that you define. In f( ), call g( ), catch its exception and, in the catch clause, throw a different exception (of a second type that you define). Test these methods from and within main( ).

Solution

```
class UDException extends RuntimeException{
public UDException(String e){
super(e);
}
}
class A{
public void f(){
try{
System.out.println("Function F");
this.g();
}catch(RuntimeException e){
System.out.println(e);
}
}
public void g() {
System.out.println("Function G");
throw new UDException("User Defined Exception");
}
}
public class Ques7 {
public static void main(String[] args) {
A alpha = new A();
alpha.f();
}
}
```

Output

```
PS F:\suvo\iem 5th sem\oops lab\a 6> cd "f:\suvo\iem 5th sem\oops lab\a 6\" ; if ($?) { javac Ques7.java } ; if ($?) { java Ques7 }
Function F
Function G
UDException: User Defined Exception
```

7. Write a program that takes one string and two integers as command line argument and prints the reverse of the substring of the string specified by the two numbers. The program should handle all possible exception that may arise due to bad input.

Solution

```
public class Ques8 {
   public static void main(String[] args) {
      String string = args[0];
      int startIndex = Integer.parseInt(args[1]);
      int endIndex = Integer.parseInt(args[2]);
      int length = string.length();
      try{
         if(startIndex<0 || endIndex>=length || endIndex<0 || startIndex>=length){
            throw new StringIndexOutOfBoundsException();
         } else if (startIndex>endIndex){
            throw new StringIndexOutOfBoundsException("StartIndex Cannot Be Greater Than
EndIndex");
         }
         System.out.println(string.substring(startIndex, endIndex));
      }catch(StringIndexOutOfBoundsException e){
         System.out.println(e);
      }
   }
}
```

Output

```
PS F:\suvo\iem 5th sem\oops lab\a 6> cd "f:\suvo\iem 5th sem\oops lab\a 6\" ; if ($?) { javac Ques8.java } ; if ($?) { java Ques8 }
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 0
        at Ques8.main(Ques8.java:3)
```