# LAB ASSIGNMENT-3

Q1. Write a program to read your lucky no. from keyboard. Treat -ve no. as NumberFormatException. Write an appropriate Exceptional Handler.

Ans.

```
import java.util.*;
public class Q1
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter your lucky no.");
        int n = sc.nextInt();
        try
        {
            if (n < 0)
            {
                throw new NumberFormatException
                ("Negetive no.");
            }
            else
            {
                System.out.println("Your lucky no is :"
                + n);
            }
        }
    }
```

Name:_____ Regd. Number:_____

```
        catch (NumberFormatException e)
        {
          System.out.println(e);
        }
        sc.close();
    }
}
```

Output:

- Enter your lucky no: 7
  Your lucky no. is 7.

- Enter your lucky no.: -10
  java.lang.NumberFormatException: Negetive No.

Q2. Assign your favourite colors in an array. Identify 2 exceptions that may be generated and write the exceptional handler in Java. Also, display the 4 colours after handling any 2 exceptions.

Ans.

```
import java.util.*;
public class Q2
{
  public static void main (String[] args)
  {
    Scanner sc=new Scanner (System.in);
    String[] col = new String[4];
    System.out.println("Enter 4 colours:");
    for (int i=0; i< col.length; i++)
    {
      col[i]=sc.next();
    }
    try
    {
      Integer.parseInt(col[0]);
    }
    try
    {
      System.out.println("Enter 1 more
        colour");
      col[5]=sc.next();
    }
```

```
        catch (NumberFormatException e)
        {
            System.out.println(e);
        }
        catch (ArrayIndexOutOfBoundException e)
        {
            System.out.println(e);
        }
        System.out.println("The colours entered
        are");
        for (int i=0; i< col.length; i++)
        {
            System.out.println(col[i]);
        }
        sc.close();
    }
}
```

Output: Enter 4 colours:
red
blue
yellow
green
java.lang.NumberFormatException: For string: "red"
Enter 1 more colour:
purple
java.lang.ArrayIndexOutOfBoundException.

Q3. Create a class Student having two instance variable name and mark. Enter mark, name of the student. If mark is more than 100; create exception MarksOutOfBoundException and throw it using Java.Displays. The customized message can't be greater than 100 for exception.

Ans.

```
import java.util.*;
class MarksOutOfBoundException extends Exception
{
    MarksOutOfBoundException(String message)
    {
        super(message);
    }
}
class Student
{
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter name and marks
    of student:");
    String name=sc.nextLine();
    try
    {
        double mark=sc.nextDouble();
        if(mark>100)
        {
            throw new MarksOutOfBoundException(
            "Can't be greater than 100");
        }
```

Name:_____ Regd. Number:_____

```
            else
            {
              System.out.println(name+" has got "+
              mark);
            }
        }
        catch (MarkOutOfBoundException e)
        {
            System.out.println(e);
        }
    }
}
```

Output :-

Enter the ~~name of the~~ name and mark of the student :

Rahul

130

MarkOutOfBoundException : ~~mark~~ can't be greater than 100.

Q4. Write a simple main class in Java that contains an experiment that uses the generics Box<T> class to build boxes with different types and that verifies that this class works as advertised. Your experiment should include the following :-

- Create a boxed String object and two variables that refer to that box. Change the contents of one and determine the effect on the other.
- Create a boxed integer object and two variables that refer to that box. Change the contents
- Create a boxed Object object and two variables that refer to that box. Determine what happens if you put an integer in the box.

Ans.

```
class Box <T>
{
   T content;
    Box (T content)
    {
       this . content = content;
    }
    T getContent()
    {
       return content;
    }
    void setcontent (T content)
    {
       this. content = content;
    }
```

```
}
public class Q4
{
    public static void main (String [] args )
    {
        Box <String> s1 = new Box<>(" It is the object
        of the Box String");
        Box <String> x1 = s1;
        Box <String> x2 = s1;
        x1. setcontent (" Modified content");
        System.out. println (" Variable 2 content :" +
        x2.getcontent());
        System.out. println (" Variable 1 content :" +
        x1. get Content());
        Box <Integer> i1 = new Box <> (6738);
        Box <Integer> y1 = i1;
        Box <Integer> y2 = i1;
        System. out. println (" Integer variable 1
        contain " + y1. getcontent());
        System. out. println (" Integer variable 2
        contain " + y2. get Content());
        y1. setcontent (99);
        System. out. println (" Now, Integer variable 1
        contain " + y1. get content());
        System. out. println (" Now, Integer variable 2
        contain " + y2. getcontent());
```

```
Box <Object> ob1 = new Box< >(new Object());
Box <Object> ob2 = ob1;
System.out.println("Object variable 1
contain" + ob1.getContent());
System.out.println("Object variable 2 contain
" + ob2.getcontent());
ob1.setContent("String inserted in box");
System.out.println("Object variable 1 contain
"+ obj.getcontent());
System.out.println("Object variable 2 contain"
+ ob2.getcontent());
    }
}
```

Output:

Variable 2 content: Modified content Modified content
variable 1 content : ~~It is the object of Box String~~
Integer variable 1 contain 6738
Integer variable 2 contain 6738
Now, integer variable 1 contain 99
Now, integer variable 2 contain 99
Object variable 1 contain java.lang.Object
Object variable 2 contain java.lang.Object.
Object variable 1 contain String inserted in box
Object variable 2 contain String inserted in box

Q5: Write a java program to print an array of different type using a single generic method. The signature of printArray method is: public static <E> void printArray(E inputArray[]).

Ans.

```
import java.util.*;
public class Q5
{
  public static void main(String[] args)
  {
    Scanner sc=new Scanner(System.in);
    System.out.println("Enter array element:");
    int a[]=new int[5];
    for(int i=0;i<a.length;i++)
    {
        a[i]=sc.nextInt();
    }
    System.out.println("Integer array contain");
    printArray(a);
    double b[]=new double[5];
    for(int i=0;i<b.length;i++)
    {
        b[i]=sc.nextDouble();
    }
    System.out.println("Double array contain");
```

Name:_____ Regd. Number:_____

```
        printArray(b);
        sc.close();
    }
    public static <E> void printArray( E array[])
    {
        for (int i=0 ; i< array.length; i++)
        {
            System.out.print(array[i] + " ");
        }
        System.out.println();
    }
}
```

Output :

```
Enter array element :
1
2
3
4
5
Integer array contains :
1 2 3 4 5

Enter array element :
1.1
2.2
3.3
4.4
5.5
Double array contain
1.1 2.2 3.3 4.4 5.5
```

~~Q5: Write a java program to print an array of different type using a single Generic method. The signature of printArray method is given below.~~

~~public static <E> void prn~~

Q6: Write a java method using Generics to count the occurance of an element in an array of any type. The signature of count method is given below..

public static int count(T array[], T item).

Ans.

```
public class Q6
{
  public static void main(String[] args)
  {
    int a[] = {1, 2, 3, 4, 1, 1, 1};
    int i = 1;
    System.out.println("Occurance of "+i+" is "
    + count(a, i));
  }
  public static <T> int count(T array[], T item)
  {
    int c=0;
    System.out.print("[");
    for(int i=0; i<array.length; i++)
    {
      if(array[i]==item)
      {
        c++;
      }
```

```
            System.out.print(array[i] + " ");
        }
        System.out.println("]");
        return c;
    }
}
```

Output:

[ 1 2 3 4 1 1 1 ]

Occurance of 1 is 4.

Q7: Write a recursive method in java that computes the factorial of a given integer.

Ans.

```
import java.util.*;
public class Q7
{
  public static void main(String[] args)
  {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a no.");
    int n = sc.nextInt();
    System.out.println("Factorial is "+fact(n));
  }
  public static int fact(int n)
  {
    if(n==0 || n==1)
    {
      return 1;
    }
    else
    {
      return n*fact(n-1);
    }
  }
}
```

Output:

```
Enter a no.5
Factorial is 120.
```

Q8. Write a recursive method in Java which given real value x and positive integer n; return the value of $x^n$.

Ans.

```
import java.util.*;
Public class Q8
{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the base : ");
      int base = sc.nextInt();
      System.out.println("Enter power : ");
      int pow = sc.nextInt();
      System.out.println(base + " to the power " +
      pow + " is " + ~~pow~~ power(base, pow));
   }
   public static int pow (int base, int pow)
   {
      if (base == 0)
      {
         return 0;
      }
      else if (power == 0)
      {
         return 1;
      }
      else
      {
         return base * power(base, power-1);
```

Name:_____ Regd. Number:_____

```
    }
  }
}
```

Output:

```
Enter base:
2
Enter power:
10
2 to the power 10 is 1024.
```

Q9 : Write a recursive method in java which, given an integer n, print it with its digits reversed. For ex. given 4735, it prints 5374.

Ans.

```
import java.util.*;
public class Q9
{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner(System.in);
      System.out.print("Enter a no.");
      int n = sc.nextInt();
      System.out.println("Reverse " + reverse(n));
   }
   public static int reverse(int n)
   {
      if (n==0)
      {
         return 0;
      }
      else
      {
         return n%10 + reverse(n/10);
      }
   }
}
```

Output:

```
Enter a no. 2024
Reverse 4202.
```

Q10: The sequence of no.s 1, 1, 2, 3, 5, 8, 13 etc are called Fibonacci no.s, each is the sum of the preceding two. Write a recursive method in Java which, given n, return nth Fibonacci no.

Ans.

```
import java.util.*;
public class Q10
{
  public static void main (String[] args)
  {
     Scanner sc=new Scanner(System.in);
     System.out.println ("Enter no. of terms :");
     int n=sc.nextInt();
     for(int i=0; i<n; i++)
     {
        System.out.print(Fibonacci(i) + " ");
     }
  }
  public static int Fibonacci (int n)
  {
     if (n==0 || n==1)
     {
        return n;
     }
     else
     {
        return fibonacci(n-1) + fibonacci(n-2);
     }
  }
}
```

Name:_____ Regd. Number:_____