



Lab Assignment 3

CSE 2001: Data Structure & Algorithms

Solution of

Programming Assignment-III

(Exception, Generics, & Recursion)

**Question-1:**

Write a Java program to read your lucky number from keyboard. Treat **One** no. as **NumberFormatException**. Write appropriate **Exceptional** handler.

CODE:-

```
import java.util.*;
public class Q1
{
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter your lucky number: ");
        int luckyno = obj.nextInt();
        try {
            if (luckyno < 0)
            {
                throw new NumberFormatException("Negative number");
            }
            System.out.println("Your lucky number is: " + luckyno);
        }
        catch (NumberFormatException e)
        {
            System.out.println(e);
        }
        obj.close();
    }
}
```

OUTPUT:-

```
//sample run-1
Enter your lucky number: -90
java.lang.NumberFormatException: Negative number
//sample run-2
Enter your lucky number: 34
Your lucky number is: 34
```

**Question-2:**

Assign your favourite colours **in** an array. Identify **2** exceptions that may be generated & write exceptional handler **in Java**. Also, display the four colours **after** handling any **2** exceptions.

CODE:-

```
import java.util.*;
public class Q2
{
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        String[] colours = new String[4];
        System.out.println("Enter four colours:");
        for (int i = 0; i < colours.length; i++)
        {
            colours[i] = obj.nextLine();
        }
        try
        {
            Integer.parseInt(colours[0]);
        }
        catch(NumberFormatException e)
        {
            System.out.println(e);
        }
        try
        {
            System.out.println("Enter one more color   ");
            colours[5]=obj.nextLine();
        }
        catch (ArrayIndexOutOfBoundsException e)
        {
            System.out.println(e);
        }
        System.out.println("The colours entered are ");
        for(int i=0;i<colours.length;i++)
        System.out.println(colours[i]);
        obj.close();
    }
}
```

OUTPUT:-

Enter four colours:

RED

BLUE

YELLOW

GREEN

java.lang.NumberFormatException: For input string: "RED"

Enter one more color

VIOLET

java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 4

The colours entered are

RED

BLUE

YELLOW

GREEN

**Question-3:**

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** & **throw** it **using Java**.Display the customized message **Mark can't be greater than 100 for** the exception

CODE:-

```
import java.util.Scanner;
class MarksOutOfBoundsException extends Exception
{
    MarksOutOfBoundsException(String message)
    {
        super(message);
    }
}
class Student
{
    public static void main(String[] args) {
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter the name of the Student :");
        String name =obj.nextLine();
        System.out.println("Enter marks ");
        try{
            double mark=obj.nextInt();
            if(mark>100)
                throw new MarksOutOfBoundsException("Mark can't be greater than 100");

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

OUTPUT:-

```
//sample run-1
Enter the name of the Student :
RAMESH
Enter marks
98
RAMESH has got 98.0
//sample run-2
Enter the name of the Student :
Raju
Enter marks
130
MarksOutOfBoundsException: Mark can't be greater than 100
```

**Question-4:**

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 printArray( E[] inputArray)
```

CODE:-

```
import java.util.*;
public class Q4
{
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter array elements:");
        Integer[] intarr = new Integer[5];
        for (int i = 0; i < 5; i++)
            intarr[i] = obj.nextInt();
        System.out.print("Integer Array contains:\n");
        printArray(intarr);
        System.out.println("Enter array elements:");
        Double[] douarr = new Double[5];
        for (int i = 0; i < 5; i++)
            douarr[i] = obj.nextDouble();
        System.out.print("Double Array contains:\n");
        printArray(douarr);
        obj.close();
    }
    public static <E> void printArray(E[] inputArray)
    {
        for (int i=0;i<inputArray.length;i++)
            System.out.print(inputArray[i] + " ");
        System.out.println();
    }
}
```

OUTPUT:-

Enter array elements:

```
1  
2  
3  
4  
5
```

Integer Array contains:

```
1 2 3 4 5
```

Enter array elements:

```
1.2  
2.3  
3.4  
4.5  
5.6
```

Double Array contains:

```
1.2 2.3 3.4 4.5 5.6
```

**Question-5:**

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

```
public static <T> int count(T[] array, T item)
```

CODE:-

```
import java.util.*;
public class Q5
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        Integer arr[]={};
        System.out.println("Enter array elements ");
        for(int i=0;i<arr.length;i++)
            arr[i]=obj.nextInt();
        System.out.println("Enter the element to search ");
        int k=obj.nextInt();
        int c=count(arr, k);
        System.out.println("Number of times "+k+" present in the array is "+c);
        obj.close();
    }
    public static <T> int count(T[] array,T item)
    {
        int count =0;
        for(int i=0;i<array.length;i++)
            if (array[i].equals(item)) // equals() method instead of array[i]==k bcz
                // incase it will String type then it will show error
                count++;
        return count;
    }
}
```

OUTPUT:-

```
Enter array elements
```

```
1  
2  
3  
4  
5
```

```
Enter the element to search
```

```
4
```

```
Number of times 4 present in the array is 1
```



Question-6:

Write a simple main `class` in Java that contains an experiment that uses the generic `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 of one and determine the effect on the other.

Create a boxed Object object and two variables that refer to that box. Determine what happens if you put a string in the box. Determine what happens if you put an integer in the box.

CODE:-

```
class Box<T>
{
    public T t;
    public Box(T t)
    {
        this.t = t;
    }
    public T get()
    {
        return t;
    }
    public void set(T t)
    {
        this.t = t;
    }
}
public class Q6 {
    public static void main(String[] args)
    {
        Box<String> strobj1 = new Box<>("It is the object of Box String");
        Box<String> strobj2 = strobj1;
        System.out.println("String variable_1 contains: " + strobj1.get());
        System.out.println("String variable_2 contains: " + strobj2.get());
        strobj1.set("Changed");
        System.out.println("String variable_1 contains: " + strobj1.get());
        System.out.println("String variable_2 contains: " + strobj2.get());
        Box<Integer> intobj1 = new Box<>(9588);
        Box<Integer> intobj2 = intobj1;
        System.out.println("Integer variable_1 contains: " + intobj1.get());
        System.out.println("Integer variable_2 contains: " + intobj2.get());
        intobj1.set(20);
        System.out.println("Integer variable_1 contains: " + intobj1.get());
        System.out.println("Integer variable_2 contains: " + intobj2.get());
        Box<Object> object1 = new Box<>(new Object());
        Box<Object> object2 = object1;
        System.out.println("Object variable_1 contains: " + object1.get());
        System.out.println("Object variable_2 contains: " + object2.get());
        object1.set(" String putted in the box ");
        System.out.println("Object variable_1 contains: " + object1.get());
        System.out.println("Object variable_2 contains: " + object2.get());
        object1.set(10);
        System.out.println("Object variable_1 contains: " + object1.get());
        System.out.println("Object variable_2 contains: " + object2.get());
    }
}
```

**OUTPUT:-**

```
String variable_1 contains: It is the object of Box String
String variable_2 contains: It is the object of Box String
String variable_1 contains: Changed
String variable_2 contains: Changed
Integer variable_1 contains: 9588
Integer variable_2 contains: 9588
Integer variable_1 contains: 20
Integer variable_2 contains: 20
Object variable_1 contains: java.lang.Object@2c7b84de
Object variable_2 contains: java.lang.Object@2c7b84de
object variable_1 contains: String putted in the box
object variable_2 contains: String putted in the box
Object variable_1 contains: 10
Object variable_2 contains: 10
```

Question-7:

Write a menu driven program to perform several mathematical operations. Different choices for the mathematical operations are as follows.

1. Determine the factorial of a number
2. Determine X^N for two numbers X, N
3. Determine GCD of two number.
4. Binary equivalent of a decimal number
5. Product of two numbers.

NOTE: All the mathematical operations must be performed using the recursive method.

CODE:-

```
import java.util.Scanner;

public class Q7
{
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        int choice;
        do
        {
            System.out.println("Choose a Mathematical operation from below :");
            System.out.println("Type 1 to Determine the factorial of a number");
            System.out.println("Type 2 to Determine  $X^N$  for two numbers X, N");
            System.out.println("Type 3 to Determine GCD of two numbers");
            System.out.println("Type 4 to Binary equivalent of a decimal number");
            System.out.println("Type 5 for Product of two numbers");
            System.out.print("Type 0 For Exit\nEnter your choice:");
            choice = obj.nextInt();
            switch (choice)
            {
                case 1:
                    System.out.print("Enter a number: ");
                    int f = obj.nextInt();
                    System.out.println("Factorial of "+f+" is "+calc_Fact(f));
                    break;
                case 2:
                    System.out.print("Enter X and N : ");
                    int x = obj.nextInt();
                    int n = obj.nextInt();
                    System.out.println( "X^N = "+calc_Powerxn(x, n));
                    break;
                case 3:
                    System.out.print("Enter first number: ");
                    int n1 = obj.nextInt();
                    System.out.print("Enter second number: ");
                    int n2 = obj.nextInt();
                    System.out.println("GCD of " + n1 + " and " + n2 + " is " + calc_Gcd(n1, n2));
                    break;
            }
        } while (choice != 0);
    }
}
```

```

        case 4:
            System.out.print("Enter a decimal number: ");
            int decimal = obj.nextInt();
            String binary = decimalToBinary(decimal);
            System.out.println(decimal + " in binary is " + binary);
            break;
        case 5:
            System.out.print("Enter first number: ");
            int p = obj.nextInt();
            System.out.print("Enter second number: ");
            int q = obj.nextInt();
            System.out.print(p + " * " + q + " = " );
            System.out.print((p<0&q<0)?-1*calc_Product(Math.abs(p),
                Math.abs(q)):calc_Product(Math.abs(p), Math.abs(q)));
            break;
        case 0:
            System.out.println("Thank You...");
            break;
        default:
            System.out.println("Invalid choice");
    }
    System.out.println();
}while (choice != 0);
obj.close();
}
public static int calc_Fact(int n)
{
    if (n == 0)
        return 1;
    else
        return (n*calc_Fact(n - 1));
}
public static int calc_Powerxn(int x, int n)
{
    if (n == 0)
        return 1;
    else
        return (x*calc_Powerxn(x, n - 1));
}
public static int calc_Gcd(int a, int b)
{
    if (b == 0)
        return a ;
    else
        return calc_Gcd(b, a % b);
}
public static String decimalToBinary(int n)
{
    if (n == 0)
        return "0";
    else if (n == 1)
        return "1";
    else
        return decimalToBinary(n / 2) + String.valueOf(n % 2);

}
public static int calc_Product(int x,int y)
{
    if (y!=0)
        return (x+ calc_Product(x, y-1));
    else
        return 0;
}
}

```

OUTPUT:-

Choose a Mathematical operation from below :
Type 1 to Determine the factorial of a number
Type 2 to Determine X^N for two numbers X, N
Type 3 to Determine GCD of two numbers
Type 4 to Binary equivalent of a decimal number
Type 5 for Product of two numbers
Type 0 For Exit

//for choice 1

Enter your choice: 1

Enter a number: 5

Factorial of 5 is 120

//for choice 2

Enter your choice:2

Enter X and N : 5 3

$X^N = 125$

//for choice 3

Enter your choice:3

Enter first number: 14 96

Enter second number: GCD of 14 and 96 is 2

//for choice 4

Enter your choice:4

Enter a decimal number: 15

15 in binary is 1111

//for choice 5

Enter your choice:5

Enter first number: 10 8

Enter second number: $10 * 8 = 80$

//for choice 0

Enter your choice:0

Thank You...

**Question-8:**

Write a recursive method in Java which, given an integer n, print it with its digits reversed. For example, given 4735, it prints 5374

CODE:-

```
import java.util.Scanner;

public class Q8
{
    static int ans=0;
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter the number that you want to reverse: ");
        int n = obj.nextInt();
        System.out.print("The reverse of the given number is: ");
        reverseNumber(n);
        obj.close();
    }

    public static void reverseNumber(int num)
    {
        if (num < 10)
        {
            System.out.print(num);
            return;
        }
        else
            System.out.print(num%10);
        reverseNumber(num/10);
    }
}

//           or you can use method
//       public static int reverseNumber(int num)
//       {
//           if (num==0)
//               return ans;
//           else
//           {
//               int r=num%10;
//               ans=ans*10 +r;
//               return reverseNumber(num/10);
//           }
//           return ans;
//       }
// }
```

OUTPIUT-:

Enter the number that you want to reverse: 3456

The reverse of the given number is: 6543

**Question-9:**

The sequence of numbers 1, 1, 2, 3, 5, 8, 13 etc are called **Fibonacci numbers**, each is the sum of the preceding two. Write a recursive method in Java which, given n, returns the nth Fibonacci number.

CODE:-

```
import java.util.*;
public class Q9
{
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter one number");
        int n = obj.nextInt();
        System.out.println("Fibonacci number at position " + n + " is " + fibonacci(n));
    }
    public static int fibonacci(int n)
    {
        if (n<=1)
            return n;
        else
            return (fibonacci(n-1)+fibonacci(n-2));
    }
}
```

OUTPIUT-:

```
Enter one number
10
Fibonacci number at position 10 is 55
```

**Question-10:**

Write a recursive Java method that takes a character string s and outputs its reverse. For example, the reverse of ' pots&pans' would be ' snap&stop'.

CODE:-

```
import java.util.Scanner;

public class Q10
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a string:");
        String s = sc.nextLine();
        String reversed = reverse(s);
        System.out.println("Reversed string: " + reversed);

    }
    public static String reverse(String s) {
        if (s.isEmpty())
            return s;

        return reverse(s.substring(1)) + s.charAt(0);
    }
}
```

OUTPIUT-:

```
//sample run-1
Enter a string:
pots&pans
Reversed string: snap&stop
//sample run-2
Enter a string:
snap&stop
Reversed string: pots&pans
```

**Home Assignment Question-1:-**

Create a class Bank with instance variables account_no, name, and balance of the customer. If the input balance is less than or equal to zero then create an Exception called "Invalid BalanceException" and throw it using Java. Display the custom message "Balance cannot be less than 0".

CODE:-

```
import java.util.*;
class InvalidBalanceException extends Exception
{
    InvalidBalanceException(String message)
    {
        super (message);
    }
}
class Bank
{
    long account_no;
    String name;
    double balance;
    public Bank(int account_no, String name, double balance) throws InvalidBalanceException
    {
        if (balance < 0)
        {
            throw new InvalidBalanceException("Balance cannot be less than 0");
        }
        this.account_no = account_no;
        this.name = name;
        this.balance = balance;
    }

    public void withdraw(double amount) throws InvalidBalanceException
    {
        if (amount > balance) {
            throw new InvalidBalanceException ("Balance cannot be less than 0");
        }
        balance -= amount;
    }
    public long getAccountNo() {
        return account_no;
    }

    public String getName() {
        return name;
    }

    public double getBalance() {
        return balance;
    }
}
```



```
public class HW1 {
    public static void main(String[] args)
    {
        Scanner obj = new Scanner(System.in);
        System.out.println("Enter name:");
        String name = obj.nextLine();
        System.out.println("Enter account number:");
        int account_no = obj.nextInt();
        System.out.println("Enter balance:");
        double balance = obj.nextDouble();
        try{
            Bank bank = new Bank(account_no, name, balance);
            System.out.println("Details of the Account Holder:");
            System.out.println("Name: " + bank.getName());
            System.out.println("Account number: " + bank.getAccountNo());
            System.out.println("Balance: " + bank.getBalance());
            System.out.println("Enter the money to withdraw:");
            double amount = obj.nextDouble();
            bank.withdraw(amount);
            System.out.println("After withdraw, Balance = " + bank.getBalance());
        } catch (InvalidBalanceException e)
        {
            System.out.println(e);
        }
    }
}
```

OUTPUT:-

//sample run-1

```
Enter name:  
Rahul  
Enter account number:  
1235  
Enter balance:  
700
```

Details of the Account Holder:

```
Name: Rahul  
Account number: 1235  
Balance: 700.0  
Enter the money to withdraw:  
300
```

After withdraw, Balance = 400.0

//sample run-2

```
Enter name:  
Rahul  
Enter account number:  
1235  
Enter balance:  
700
```

Details of the Account Holder:

```
Name: Rahul  
Account number: 1235  
Balance: 700.0  
Enter the money to withdraw:  
900
```

InvalidBalanceException: Balance cannot be less than 0

**Home Assignment Question-2:**

Write a recursive method in Java to search an element of an array using binary search.

CODE:-

```
import java.util.*;
public class HW2
{
    public static int binarySearch(int arr[], int start, int end, int key)
    {
        if (end >= start && start <= arr.length - 1)
        {
            int mid = start + (end - start) / 2;
            if (arr[mid] == key)
                return mid;
            if (arr[mid] > key)
                return binarySearch(arr, start, mid - 1, key);

            return binarySearch(arr, mid + 1, end, key);
        }
        return -1;
    }
    public static void main(String args[])
    {
        Scanner obj=new Scanner(System.in);
        System.out.println("Enter number of elements in the array");
        int n=obj.nextInt();
        int arr[] = new int[n];
        System.out.println("Enter "+n+" number of elements in ascending order");
        for (int i=0;i<n;i++)
            arr[i]=obj.nextInt();
        System.out.println("Enter the element to search");
        int x = obj.nextInt();
        int result = binarySearch(arr, 0, arr.length-1, x);
        System.out.println(result== -1?("The "+x+" is not present in the array"):(("The "+x+" is present at index "+result)));
    }
}
OUTPUT:- Enter number of elements in the array
5
Enter 5 number of elements in ascending order
11
22
33
44
55
Enter the element to search
33
The 33 is present at index 2
```

**Home Assignment Question-3:**

Write a recursive Java method that determines if a string s is a palindrome, that is, it is equal to its reverse.

Examples of palindromes include 'racecar' and 'gohangasalamimalasagnahog'.

CODE:-

```
import java.util.*;
import java.util.*;
public class HW3
{
    static boolean isPalindrome(String str,int s, int e)
    {
        if (s == e)
            return true;
        if ((str.charAt(s)) != (str.charAt(e)))
            return false;
        if (s < e + 1)
            return isPalindrome(str, s + 1, e - 1);
        return true;
    }
    public static void main(String args[])
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter the String : ");
        String str = obj.nextLine();
        System.out.println(isPalindrome(str,0,str.length()-1)?"Yes it is a pallindrome String.":"No
it is not a pallindrome String.");
    }
}
```

OUTPIUT-:

//sample run-1

Enter the String : racecar

Yes it is a pallindrome String.

//sample run-2

Enter the String : gohangasalamimalasagnahog

Yes it is a pallindrome String.



Home Assignment Question-4:

Given an unsorted array, A, of integers and an integer k, write recursive program using Java for rearranging the elements in A so that all elements less than or equal to k come before any elements larger than k.

CODE:-

```
import java.util.*;
public class HW4
{
    public static void main(String[] args)
    {
        Scanner obj=new Scanner(System.in);
        System.out.print("Enter the length of the array : ");
        int n=obj.nextInt();
        int[] A = new int[n];
        System.out.println("Enter the values in the array : ");
        for(int i=0;i<n;i++)
            A[i]=obj.nextInt();
        System.out.print("Enter k : ");
        int k = obj.nextInt();
        rearrangeArray(A, k);
        System.out.println("After rearrangement the array :");
        for(int i=0;i<n;i++)
            System.out.print(A[i]+" ");
    }
    public static void rearrangeArray(int[] A, int k)
    {
        halfQuicksort(A, k, 0, A.length - 1);
    }
    public static void halfQuicksort(int[] A, int k, int start, int end)
    {
        if (start >= end)
            return;
        int cutIndex = getPivot(A, k, start, end);
        halfQuicksort(A, k, start, cutIndex - 1);
        halfQuicksort(A, k, cutIndex + 1, end);
    }
}
```



//Written by Ansuman Swain

```
public static int getPivot(int[] A, int k, int start, int end) {  
    int pivot = A[end];  
    int i = start - 1;  
    for (int j = start; j <= end - 1; j++)  
        if (A[j] <= k)  
        {  
            i++;  
            swap(A, i, j);  
        }  
    swap(A, i + 1, end);  
    return i + 1;  
}  
public static void swap(int[] A, int i, int j)  
{  
    int temp = A[i];  
    A[i] = A[j];  
    A[j] = temp;  
}
```

}

OUTPIUT-:

the length of the array : 10

Enter the values in the array :

5 9 6 16 18 3 1 2 7 22

Enter k : 5

After rearrangement the array :

5 3 1 2 22 18 9 6 16 7

**Home Assignment Question-5:**

In the Towers of Hanoi puzzle, we are given a platform with three pegs, a, b, and c, sticking out of it. On peg a is a stack of n disks, each larger than the next, so that the smallest is on the top and the largest is on the bottom. The puzzle is to move all the disks from peg a to peg c, moving one disk at a time, so that we never place a larger disk on top of a smaller one. See Figure 1 for an example of the case n = 4. Write a recursive program using Java for solving the Towers of Hanoi puzzle for arbitrary n. (Hint: Consider first the sub problem of moving all but the nth disk from peg a to another peg using the third as “temporary storage.”)

CODE:-

```
public class HW5 {

    public static void towerOfHanoi(int n, char first_rod,char last_rod, char middle_rod)
    {
        if (n == 0)
            return;
        towerOfHanoi(n - 1, first_rod, middle_rod, last_rod);
        System.out.println("Move disk " + n + " from rod "+ first_rod + " to rod " + last_rod);
        towerOfHanoi(n - 1, middle_rod, last_rod, first_rod);
    }
    public static void main(String args[])
    {
        int N = 4;
        towerOfHanoi(N, 'A', 'C','B');
    }
}
```

OUTPUT:-

```
Move disk 1 from rod A to rod B
Move disk 2 from rod A to rod C
Move disk 1 from rod B to rod C
Move disk 3 from rod A to rod B
Move disk 1 from rod C to rod A
Move disk 2 from rod C to rod B
Move disk 1 from rod A to rod B
Move disk 4 from rod A to rod C
Move disk 1 from rod B to rod C
Move disk 2 from rod B to rod A
Move disk 1 from rod C to rod A
Move disk 3 from rod B to rod C
Move disk 1 from rod A to rod B
Move disk 2 from rod A to rod C
Move disk 1 from rod B to rod C
```