Department of Computer Science & Engineering, SDMCET, Dharwad-2

[Submitted as part of CTA Assignment No-1]

Course:	Advanced Object-Oriented Programming	Course Code:	18UCSE508
Semester:	V	Division:	B

Submitted by:

USN:	2SD20CS121	Name:	VENUGOPAL K MALLI

1. Problem definition:

Write a Java program to generate and handle any three built-in exceptions and display appropriate
error messages.

2. Java Program:

```
class aoop1 {
   public static void main(String[] args) {
   try {
         int a = 30, b = 0;
         int c = a / b; // cannot divide by zero
         System.out.println("Result = " + c);
 }
   catch (ArithmeticException e) {
          System.out.println("Divide by 0 error");
   }
   try {
         int a[] = new int[3];
          a[4] = 9;
   }
   catch (ArrayIndexOutOfBoundsException e) {
         System.out.println(" over limit ");
   }
   try {
         String a = null; // null value
         System.out.println(a.charAt(0));
   }
         catch (NullPointerException e) {
         System.out.println("NullPointerException");
   }
 }
}
```

3. Output:


PROBLEMS 45
OUTPUT
TERMINAL
DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\venu\codes> cd "c:\venu\codes\java\" ; if ($?) { javac aoop1.java } ; if ($?) { java aoop1 }
Divide by 0 error
over limit
NullPointerException
PS C:\venu\codes\java>

1. Problem definition:

Write a Java program to read an integer and check whether the number is prime or not. If negative number is entered, throw an exception NegativeNumberNotAllowedException and if entered
number is not prime, then throw NumberNotPrimeException.

2. Java Program:

```
import java.util.Scanner;

class negativeNum extends Exception{
   int n;
   negativeNum(int n){
      this.n=n;
   }
   public String toString(){
      return "The number"+this.n+" is  negative";
   }
}

class notPrime extends Exception{
   int n;
   notPrime(int n){
      this.n=n;
   }
   public String toString(){
      return "The number"+ this.n+" is  not prime";
   }
}

class aoop2 {
   public static void main(String [] args) throws Exception{
      Scanner sc =new Scanner(System.in);
      System.out.println("Enter the number");
      int a=sc.nextInt();
      if(a<0){
         throw new negativeNum(a);
      }
      else if(a==0||a==1){
         System.out.println(a+" is Not prime");
         throw new notPrime(a);
      }
      else{
```

```
            for(int i=2;i<a;i++){
                if(a%i!=0){
                    continue;
                }
                else{
                    System.out.println(a+" is Not prime");
                    throw new notPrime(a);

                }


            }
            System.out.println(a+" is prime");
        }
    }
}
```

3. Output:

```
Enter the number
2
2 is prime
PS C:\venu\codes\java> cd "c:\venu\codes\java\" ; if ($?) { javac aoop2.java } ; if ($?) { java aoop2 }
Enter the number
11
11 is prime
PS C:\venu\codes\java> cd "c:\venu\codes\java\" ; if ($?) { javac aoop2.java } ; if ($?) { java aoop2 }
Enter the number
4
4 is Not prime
Exception in thread "main" The number4 is  not prime
        at aoop2.main(aoop2.java:43)
PS C:\venu\codes\java> cd "c:\venu\codes\java\" ; if ($?) { javac aoop2.java } ; if ($?) { java aoop2 }
Enter the number
-1
Exception in thread "main" The number-1 is  negative
        at aoop2.main(aoop2.java:29)
PS C:\venu\codes\java>
```

1. Problem definition:

Write a Java program to perform the following operations:

a) Read a line of text
b) Search for a sub-string SDMCET (case insensitive search)
c) If found, then print success message
d) Otherwise throw an exception SubStringNotFoundException with appropriate message

2. Java Program:

```
import java.util.Scanner;

class subStringNotFound extends Exception {
   String s;
   subStringNotFound(String s){
   this.s=s;
  }
  public String toString(){
   return "The string "+this.s+" does not contain the substring";
 }
}


class aoop3 {
   public static void main(String [] args) throws Exception{
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the String");
      String s1 = sc.nextLine();

      String s2 = "SDMCET";

      if(s1.contains(s2)){
         System.out.println(s1+" Contains the the sub string SDMCET");
      }
      else{
         throw new subStringNotFound(s1);
      }

   }
}
```

3. Output:

```
PS C:\venu\codes\java> cd "c:\venu\codes\java\" ; if ($?) { javac aoop3.java } ; if ($?) { java aoop3 }
Enter the String
HII SDMCET
HII SDMCET Contains the the sub string SDMCET
PS C:\venu\codes\java> cd "c:\venu\codes\java\" ; if ($?) { javac aoop3.java } ; if ($?) { java aoop3 }
Enter the String
HIi SdmCET
Exception in thread "main" The string HIi SdmCET does not contain the substring
        at aoop3.main(aoop3.java:26)
PS C:\venu\codes\java>
```

1. Problem definition:

Write a Java program to perform the following operations:

a) Create a file named Alphabets.txt and insert appropriate data into it
b) Read the file and copy all the consonants into another file named Consonants.txt
c) If vowel is encountered, throw an exception VowelNotAllowedException and continue until
end of file

2. Java Program:

```
import java.io.*;
import java.util.Scanner;
import java.io.FileInputStream;
import java.io.FileOutputStream;


class  VowelNotAllowedException extends Exception{
   char a;
   VowelNotAllowedException(char a){
      this.a=a;
   }
   public String toString(){
      return "The alphabet "+this.a+" is not allowed ";
    }
}
public class aoop4 {
   public static void main(String [] args) throws Exception{

      FileOutputStream f1=new FileOutputStream("Alphabets.txt");
      FileOutputStream fot=new FileOutputStream("Consonants.txt");
      FileInputStream fin = new FileInputStream("Alphabets.txt");
      Scanner sc = new Scanner(System.in);
      System.out.println("Enter the String");
      String s=sc.nextLine();
      String s1=s.toLowerCase();

      int l=s1.length();
      for(int i=0;i<l;i++){
         char ch=s1.charAt(i);
         f1.write(ch);
```

```
            if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'){
               ss
               continue;
            }
            else {
               fot.write(ch);
            }

          }
          int n;
         while((n=fin.read())!=-1){
          char ch =(char) fin.read();
          if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'){

             throw new VowelNotAllowedException(ch);
          }
         }

       System.out.println("Entered String contains only consonents");


      }

    }
```

3. Output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\venu\codes> cd "c:\venu\codes\java\" ; if ($?) { javac aoop4.java } ; if ($?) { java aoop4 }
Enter the String
Hello there
Exception in thread "main" The alphabet e is not allowed
        at aoop4.main(aoop4.java:47)
PS C:\venu\codes\java>
```

```
java > Consonants.txt
  1    hll thr
```

```
java > Alphabets.txt
  1    hello there
```

1. Problem definition:

Write a Java program to implement the following scenario:

a) Create a file named Integers.txt and insert n-random integers into it
b) Create three threads T1, T2 and T3 that read n/3 integers in sequence of occurrence of
numbers from the file and sort the read n/3 integers
c) Thread T4 waits for all the threads T1, T2 and T3 to complete sorting, then sorts and outputs
the entire list of sorted numbers to another file named SortedIntegers.txt

2. Java Program:

```
import java.util.*;
import java.util.Scanner;
import java.io.*;

public class Qq5 {
	public static void main(String[] args) {
		try{
			FileWriter w = new FileWriter("Integer.txt");
			Scanner sc= new Scanner(System.in);
			System.out.println("Enter the value of n Integer to
write on a file :");
			int n = sc.nextInt();
			for (int i = 0; i < n; i++) {
				System.out.print("Enter the " + (i + 1) + "to
write :" );
				int input = sc.nextInt();
				w.write(input + "\t");
			}
			w.close();
			int i=0;
			int arr[] = new int[n];
			File file = new File("Integer.txt");
			Scanner read = new Scanner(file);
			while(read.hasNext()){
				arr[i++] = Integer.valueOf(read.next());
			}
			Thread t1= new Thread(){
				public void run(){
```

```
                                        Arrays.sort(arr, 0, (arr.length/3));
                                        for (int j = 0; j < (arr.length/3); j++) {
                                                System.out.println(arr[j]);
                                        }
                                }
                        };
                        Thread t2= new Thread(){
                                public void run(){
                                        Arrays.sort(arr, (arr.length/3),
(2*(arr.length/3)));
                                        for (int j = (arr.length/3); j <
(2*(arr.length/3)); j++) {
                                                System.out.println(arr[j]);
                                        }
                                }
                        };
                        Thread t3= new Thread(){
                                public void run(){
                                        Arrays.sort(arr, (2*(arr.length/3)),(n-1));
                                        for (int j = (2*(arr.length/3)); j < n; j++) {
                                                System.out.println(arr[j]);
                                        }
                                }
                        };
                        Thread t4= new Thread(){
                                public void run(){
                                        Arrays.sort(arr);
                                        StringBuilder s = new StringBuilder();
                                        try{
                                                FileWriter write =new
FileWriter("SortedInteger.txt");
                                                System.out.println("t4 is
printing");
                                                for (int j = 0; j < n; j++) {

        s.append(String.valueOf(arr[j]) + "\t");
                                                }
                                                write.write(s.toString());
                                                write.close();
                                        }catch (Exception e){
                                                System.out.println(e);
                                        }
                                }
```

```
                    };

                    t1.start();
                    t1.join();
                    t2.start();
                    t2.join();
                    t3.start();
                    t3.join();
                    t4.start();
                }catch(Exception e){
                    System.out.println(e);
                    }
                }
}
```

3. Output:

```
PS C:\venu\codes\java> cd "c:\venu\codes\java\" ; if ($?) { javac aoop5.java } ; if ($?) { java aoop5 }
Enter the value of n Integer to write on a file :
5
Enter the 1to write :1
Enter the 2to write :3
Enter the 3to write :2
Enter the 4to write :4
Enter the 5to write :8
1
3
2
4
8
t4 is printing
```

```
java > SortedInteger.txt
  1   1   2   3   4   8
```