

LAB 01

PART 01

Task 1:

Given an array of ints, return true if 6 appears as either the first or. Last element in the array. The array will be length 1 or more.

firstLast6 ({1, 2, 6}) → true

firstLast6 ({6, 1, 2, 3}) → true

firstLast6 ({13, 6, 1, 2, 3}) → false

Task 2

Read in 10 input from the keyboard, and store them in an array. Find the position (or index) of the maximum and minimum values in the array, and swap them (move the biggest element to the position of the smallest, and move the smallest element to the position of the biggest) and print that again. If the user enters 7, 13, -5, 10, 6 then your program should print 7, -5, 13, 10, 6

Task 3

Write a program which reads 5 numbers into an array, sorts/arranges the numbers from high to low and prints all numbers in the array and the even number in that list.

If the user enters 7, 13, 2, 10, 6 then your program should print 13, 10, 7, 6, 2, 10, 6 , 2.

Task 4

Write a Java program that would ask the user to enter ten numbers. As the user enters the ten numbers the program will make sure that all the numbers entered by the user are unique. For example if the user enters 2, 3, 4, 6, and then tries to enter 3 again the program will display that 3 is already in among the entered numbers and would ask the user to enter a new number.

Task 5

Write a Java program that would ask the user to enter 15 numbers. All the numbers are between 0 and 9. The program should count how many time the number been given.

In the Input:

1,1,2,5,6,8,7,2,4,6,9,4,2,5,6

Output should look like:

0 was found 0 times

1 was found 2 times

2 was found 3 times

3 was found 0 times

4 was found 2 times

5 was found 2 times

6 was found 3 times

7 was found 1 times

8 was found 1 times

9 was found 1 times

Hint: Using an array, use the element $a[n]$ to count frequency / occurrence of the number n .

Task 6

An array is a *palindrome* if it reads the same in both directions. For example,

3 5 7 2 4 is not a palindrome; however, the following one is. 4 2 6 2 4

Write a program that reads in an array and checks if it is a palindrome

Task 7

Read 10 input from the user, and store them in an array. Then ask the user to give another input between 0 and 9 for choosing an array cell. Display a line with that many "*" characters.

Eg. Say if the user enter 15,6,7,9,45,2,26,84,8,10 then the user again give input as 3 the output will be "*****"

Task 8

Ask user for dimension/size of two row matrices, take all values in each matrix and print the summation. If matrices are A and B, then you need to calculate $C=5A-B$. For

example, if the user enters 3, then size of each matrix is 3. Then you need to take 3 values for first matrix which will form $A = \begin{bmatrix} 5 & 6 & 7 \end{bmatrix}$. Then take 3 values for second matrix, $B = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix}$. You need to print result, $C = \begin{bmatrix} 23 & 27 & 31 \end{bmatrix}$. The output will be 23 27 31. You must use array, loop (while or for), and array.length for this program.

Task 9

Write a Java Code of a program that declares an array of integer type of size 10 and takes input of your choice. Have the program print the values, square all the values, and then display the new values. Write a static method (in the same class as main method) to do the displaying and a second static method to find out the squares. Have the methods take the array name and the array size as arguments.

Task 10

Write a method named `season` that takes two integers as parameters representing a month and day and that returns a String indicating the season for that month and day. Assume that months are specified as an integer between 1 and 12 (1 for January, 2 for February, and so on) and that the day of the month is a number between 1 and 31.

If the date falls between 12/16 and 3/15, you should return "Winter". If the date falls between 3/16 and 6/15, you should return "Spring". If the date falls between 6/16 and 9/15, you should return "Summer". And if the date falls between 9/16 and 12/15, you should return "Fall".

Task 11

Write a method named `allDigitsOdd` that returns whether every digit of a positive integer is odd. Your method should return true if the number consists entirely of odd digits and false if any of its digits are even. 0, 2, 4, 6, and 8 are even digits, and 1, 3, 5, 7, 9 are odd digits.

For

example, `allDigitsOdd(135319)` returns true but `allDigitsOdd(9145293)` returns false.

Task 12

Write a method `before` that takes as parameters two month/day combinations and that returns whether or not the first date comes before the second date (true if the first month/day comes before the second month/day, false if it does not). The method will take four integers as parameters that represent the two month/day combinations.

The first integer in each pair represents the month and will be a value between 1 and 12 (1 for January, 2 for February, etc, up to 12 for December). The second integer in each

pair represents the day of the month (a value between 1 and 31). One date is considered to come before another if it comes earlier in the year.

Task 13

Write the **removeOdd** method bellow which take in an array of numbers that has even and odd numbers mixed. This function **removes** the odd numbers and returns a **compact** array which only has the even numbers. For example output of the following code is:

21 33 44 66 11 1 88 45 10 9 44 66 88 10
--

```
public class Test{
    public static int [] removeOdd (int [] input){
        //Your code here
    }
    public static void main(String [] args){
        int [] mixedArray = {21, 33, 44, 66, 11, 1, 88, 45, 10, 9};
        for (int i = 0; i < mixedArray.length; i++) {
            System.out.print(mixedArray[i] + " ");
        }
        System.out.println();
        int [] noOdd = removeOdd(mixedArray);
        for (int i = 0; i < noOdd.length; i++) {
            System.out.print(noOdd[i] + " ");
        }
    }
}
```

PART 02

Task 1

Create an array of size 5. Try to store 100 to the index 21 of the array. It should give you a runtime error. Note the first line of the error that was given.

You do not have to handle/catch anything for this task, just note the errors.

Task 2

In your main method, try to divide 1 by zero. Note the first line of the error that was given.

You do not have to handle/catch anything for this task, just note the errors.

Task 3

```
import java.util.Scanner;
public class Task3{
    public static void main(String[]args){
        Scanner sc=new Scanner(System.in);
        int x,n=sc.nextInt();
        int a[]=new int[n];
        a[5]=99;
        x=n/0;
    }
}
```

When we enter 3 (or any number less than 6), then “a[5]=99;” line will cause error. Giving 6 or greater numbers will not cause that problem. Regardless of the number, “n/0” will always cause error.

Modify the program above to handle exception ‘**ArithmeticException**’ and ‘**ArrayIndexOutOfBoundsException**’ using try, catch and finally keywords. Inside finally, just print the message “**THE END**”.

Task 4

Run your solution of Task3 and give 2.5 as input. It will cause error.

Now you see that it is hard to know every error that may occur in advance. Hence, modify your solution to Task 3 to account for all other unknown errors that can be caused from any line of the main method.

Task 5

Change and complete the MyReader class so that you can create your own exception ‘**EitaIntegerNoiException**’. There will be a method int readInteger(). This method will take a String form the user. Then it will check if the number was ‘int’ or ‘float’ by

checking presence of the decimal point in the input. It will throw an exception when a floating-point number is entered. Otherwise, it will return the integer number.

Task 6

Create a class called Square as described below:

- **Fields:**
height, width
- **Methods:**
public double getHeight()
public void setHeight(double h)
public double getWidth ()
public void setWidth (double w)
public double getArea ()

Hint: If I take your class and use it following would be the code and the output.

Code	Output
<pre>double h, w, a; Square s1 = new Square(); s1.setHeight(3); s1.setWidth(4); h = s1.getHeight(); w = s1.getWidth(); a = s1.getArea(); System.out.println("Height = "+ h); System.out.println("Width = "+ w); System.out.println("Area = "+ a);</pre>	<pre>Height = 3.0 Width = 4.0 Area = 12.0</pre>

Task 7

Create a class called BankAccount as described below:

- **Fields:**
name, address, accountID, balance

- **Methods:**

```
public String getName()
public void setName(String n)
public String getAccountID()
public void setAccountID(String i)
public String getAddress()
public void setAddress(String a)
public double getBalance()
public void setBalance(double c)
public void addInterest() //adds 7% of the balance
```

Write a class called Main to write a main() method:

- ```
public static void main(String[] args){
```
- ```
}
```
- Inside the main() method
 - Create 3 objects/instances of BankAccount called acc1, acc2 and acc3
 - Set their fields to some value using the public methods.
 - Call addInterest() on acc1 and acc3
 - Print the information of each BankAccount using System.out.println()

Add constructors to Student and BankAccount and use the constructor to set the field values.

Task 8

Create a class called Date that includes three pieces of information as instance variables a month (type int), a day (type int) and a year (type int). Your class should have a constructor that initializes the three instance variables and assumes that the values provided are correct. Provide a set and a get method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date's capabilities.

Task 9

Design a class for Quadratic equation named QuadEqn consisting of 3 coefficients, a,b,c

- Take input from user in main
- print this QuadEqn
- create a method to calculate 1st root
- print one of the roots from main. (assume all roots are real, use + instead of +/-)

For example, if user gives 1, -5, 6

then print the equation: $x^2 - 5x + 6$

and 3 (the root/solution)

Task 10

Array Left Rotate and Right Rotate: Consider the 1, 4, 8, 16, 25, 36, 49, 64, 81, 100. Now, right and left rotate the array by the no. of position given by the user. Example: If user gives '4' as no. of position to be rotated, your left rotate should give output '25, 36, 49, 64, 81, 100, 1, 4, 8, 16' and right rotate should give output '49, 64, 81, 100, 1, 4, 8, 16, 25, 36'.

Task 11

Write the **ComplexNumber** class so that the following code generates the output below:

public class Tester {
public static void main(String[] args) {
RealNumber rn = new ComplexNumber();
System.out.println(rn);
System.out.println("-----");
rn = new ComplexNumber(5, 7);
System.out.println(rn);
System.out.println("-----");
ComplexNumber cn = new ComplexNumber();
cn.check();
}
}
public class RealNumber {
private double realValue;
public double getRealValue() {
return realValue;
}
public void setRealValue(double r) {
realValue = r;
}
public RealNumber() {
this(0);
}
public RealNumber(double r) {
setRealValue(r);
}
public String toString() {
return "RealPart: "+getRealValue();
}

public void ping() {
System.out.println("I'm in RealNumber class");
}
}

RealPart: 1.0
ImaginaryPart: 1.0

RealPart: 5.0
ImaginaryPart: 7.0

I'm in ComplexNumber class
I'm in RealNumber class
Checking ended.

Task 12

Write the **CSEStudent** and **CSE111Student** class so that the following code generates the output below [Answer on the answer-script]

public class Student{
public String msg = "I love BU";
public String shout(){
return msg;
}
}
public class TestStudent{
public static void printShout(Student s){
System.out.println("-----");
System.out.println(s.msg);
System.out.println(s.shout());
}
public static void main(String [] args){
Student s = new Student();

CSEStudent cs = new CSEStudent();
CSE111Student cs111 = new CSE111Student();
System.out.println(s.msg);
System.out.println(cs.msg);
System.out.println(cs111.msg);
printShout(s);
printShout(cs);
printShout(cs111);
}
}

Output
I love BU I want to transfer to CSE I love Java Programming ----- I love BU I love BU ----- I love BU I want to transfer to CSE ----- I love BU I love Java Programming