Lab06
Simple Graphical User Interface (GUI) Programming

Submission Details

In this lab, you are required to write **THREE** Java programs to solve the given problems shown in the section “Lab Exercises”. To make the program implementation easier, you are suggested to write your program using **eclipse** instead of doing it directly using paper & pencil. After you have completed the implementation using **eclipse**, submit the program files (i.e. ArithmeticCalGUI.java, DayOfWeekGUI.java, QuadraticCalculatorGUI.java) to the assignment page of Canvas. The deadline would be **ONE** week after your lab section is conducted. You are reminded to double-check your solution to verify everything in correct before submission.

Note: You are required to put your name, student ID and lab section at the beginning of your source file.

Note: Only answer of selected question will be graded.

Objective

The objective of this lab is to reinforce the programming concepts (e.g. basic Java input / output, variable declarations and use of operators) taught in the last few lectures. Also, it also gives you an opportunity to do simple **G**raphical **U**ser **I**nterface (**GUI**) programming.

Software Required

The following software is required for this lab.

1. Java Development Kit, eg JavaSE 1.8
2. An Integrated Development Environment, e.g. eclipse

Introduction

In this lab, you will again practice and try to explore:

1. Forming a program skeleton by declaring a class with a main method / operation
2. Declaring variables to store values in Java programs
3. Obtaining inputs from keyboard using **static** methods / operations in the JOptionPane (This class is in the package javax.swing).
4. Solving problems by performing arithmetic operations (+, -, *, /, %) on inputs
5. Displaying the solution of problems on screen using **static** methods / operations in the JOptionPane.

Background

1. Form a program skeleton by defining a class with a main method / operation

- **Syntax:**

```
// The filename of the Java source file should be exactly the same
// as the <class name>
public class <class name>    {
     // Define a main method / operation, which is the starting point of the program
     public static void main(String[] args)
     {
         // Program statements here
     }
}
```

2. Declare variables for storing values

- **Syntax:**

```
// Declare a variable with name <name of variable> in type <type>
<type> <name of variable>;
// Declare a variable with name <name of variable> in type <type> with
// initial value <initial value>
<type> <name of variable> = <initial value>;
// Declare a number of variables with name <variable name1>,
// <variable name2>, … all in type <type>
<type> <variable name1>, <variable name2>, …;
// Declare a number of variables with name <variable name1>,
// <variable name2>, … all in type <type> with initial value <initial value1>,
// <initial value2>, …, respectively
<type> <variable name1> = <initial value1>, <variable name2> = <initial value2>, …;
where <type> could be in one of the following types:
byte, short, int, long, float, double, char, boolean
```

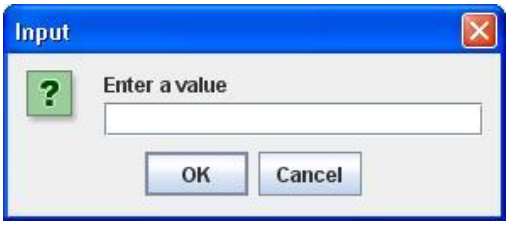
3. Obtain inputs from keyboard and store them in variables:

- **Syntax:**

```
// Get the class definition of the JOptionPane class from javax.swing package
import javax.swing.JOptionPane;
// Obtain back a String from the input dialog and convert it into double value
// using parseDouble operation provided by the Double class
String <string variable name> = JOptionPane.showInputDialog(“<text to show>”);
double val = Double.parseDouble(<string variable name>);
```

The following shows an example:

```
import javax.swing.JOptionPane;
public class SimpleGUIProgram
{
  public static void main(String[] args)
  {
    String valueStr = JOptionPane.showInputDialog("Enter a value");
    double val = Double.parseDouble(valueStr);
  }
}
```



Enter a value

4. Perform calculation using arithmetic operations:

- **Syntax:**

```
// Add the two values stored in variable 1 and variable 2
<name of variable for the result> = <name of variable 1> + <name of variable 2>;
// Subtract the value stored in variable 2 from variable 1
<name of variable for the result> = <name of variable 1> - <name of variable 2>;
// Multiply the value stored in variable 1 by the value stored in variable 2
<name of variable for the result> = <name of variable 1> * <name of variable 2>;
// Divide the value stored in variable 1 by the value stored in variable 2
<name of variable for the result> = <name of variable 1> / <name of variable 2>;
// Find the remainder of value stored in variable 1 divided by the value stored in
// variable 2
<name of variable for the result> = <name of variable 1> % <name of variable 2>;
```

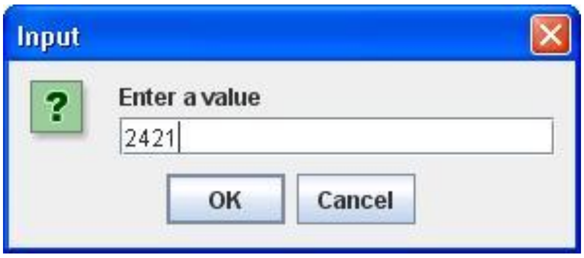
5. Display information on screen:

- **Syntax:**

// Display data in a dialog message box
String <string variable name> = “<string to show>” + <value variable name>;
JOptionPane.showMessageDialog(null, <string variable name>);

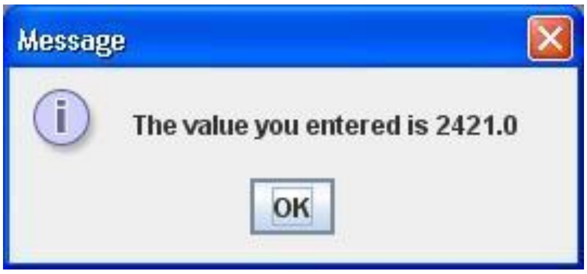
The following shows an example:

```
import javax.swing.JOptionPane;

public class SimpleGUIProgram
{
  public static void main(String[] args)
  {
    String valueStr = JOptionPane.showInputDialog("Enter a value");
    double val = Double.parseDouble(valueStr);
    String strToShow = "The value you entered is " + val;
    JOptionPane.showMessageDialog(null, strToShow);
  }
}
```



Enter a value

2421



The value you entered is 2421.0

Lab Exercises

In this lab, you will be asked to solve three problems using Java. For these three problems, write Java programs according to following problem specifications.

Question 1:

With the given Java program (see Figure 1), **MODIFY** it to use GUI dialog-based input with JOptionPane instead of console-based input using Scanner. Since operation / method showInputDialg only returns a String, you must convert the String the user enters to an int for use in calculations. Operation / Method Integer.parseInt(String s) takes a String argument representing an integer (e.g., the result of JOptionPane.showInputDialog) and returns the value as an int. If the String does not contain a valid integer, then the program will terminate with an error.

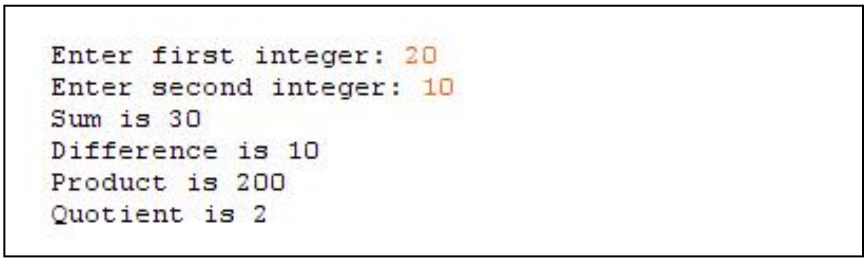
(Note: Name the modified program as ArithmeticCalGUI.java and the output of this given program is shown in Figure 2. The expected output of the modified program is shown in Figure 3.)

Original program:

ArithmeticCalculation.java

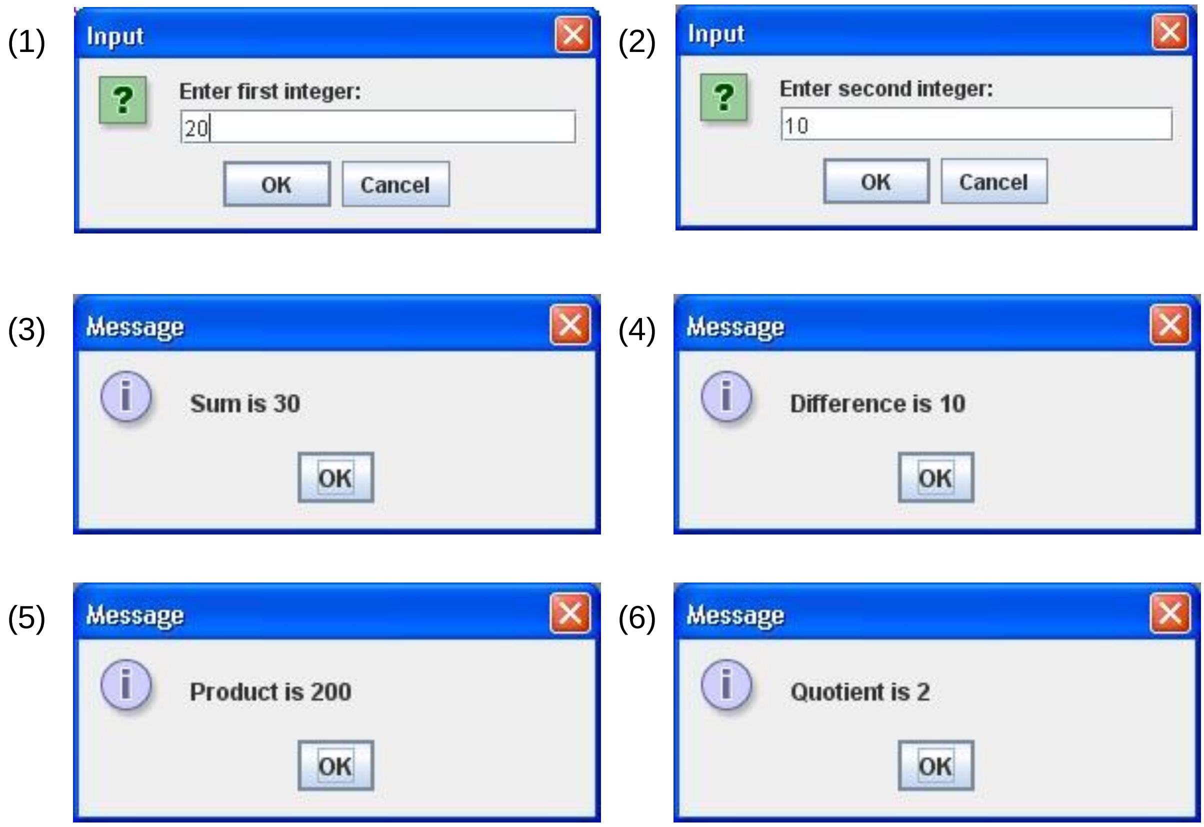
```java
import java.util.Scanner;

public class ArithmeticCalculation
{
   public static void main(String[] args)
   {
      Scanner sc = new Scanner( System.in );
      int number1;
      int number2;
      System.out.print("Enter first integer: ");
      number1 = sc.nextInt();
      System.out.print("Enter second integer: ");
      number2 = sc.nextInt();
      System.out.println("Sum is " + (number1 + number2));
      System.out.println("Difference is " + (number1 - number2));
      System.out.println("Product is " + (number1 * number2));
      System.out.println("Quotient is " + (number1 / number2));
   }
}
```

Figure 1: Sample Java program



```
Enter first integer: 20
Enter second integer: 10
Sum is 30
Difference is 10
Product is 200
Quotient is 2
```

Figure 2: Output of the original program



(1) Input

Enter first integer:

20

(2) Input

Enter second integer:

10

(3) Message

Sum is 30

(4) Message

Difference is 10

(5) Message

Product is 200

(6) Message

Quotient is 2

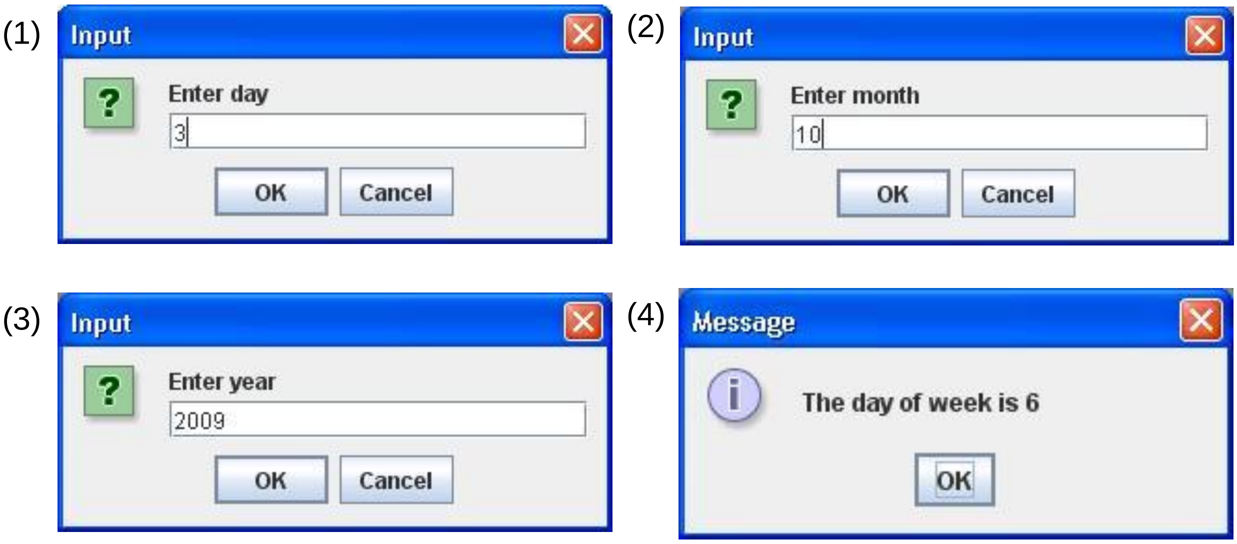
Figure 3: Expected output of the modified program (From step1 to step 6)

Question 2:

Write a GUI-based Java program that takes a date as input and prints the day of the week that the date falls on. Your programs should take three values: month, day, year. For month use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following formulas, for the Gregorian calendar:

y = year – (14 – month) / 12
x = y + y/4 – y/100 + y/400
m = month + 12 * ((14-month) / 12) – 2
d = (day + x + (31 * m) / 12) % 7

(Note: Name the program as DayOfWeekGUI.java, and the expected output of the program is shown in Figure 4.)



(1) Input
Enter day
3

(2) Input
Enter month
10

(3) Input
Enter year
2009

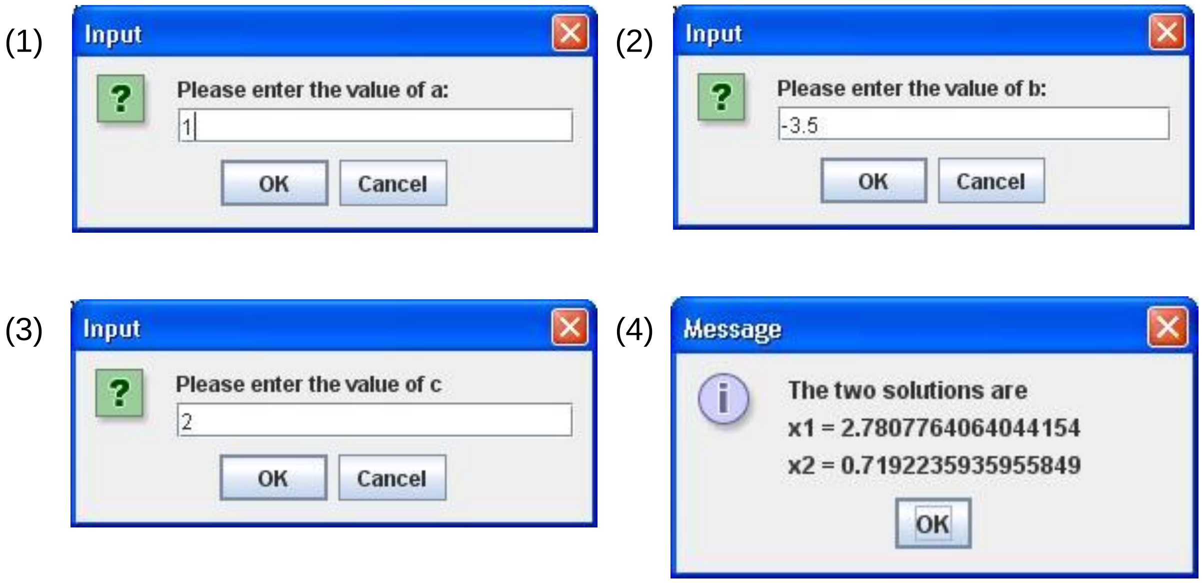
(4) Message
The day of week is 6

Figure 4: Expected output of the program (From step1 to step 4)

Question 3:

With the Java program (QuadraticCalculator.java) that you implemented in **Lab05**, **CONVERT** it to use GUI dialog-based input with JOptionPane instead of console-based input using Scanner. Since operation / method showInputDialg only returns a String, you must convert the String the user enters to a double for use in calculations. Operation / Method Double.parseDouble(String s) takes a String argument representing a double (e.g., the result of JOptionPane.showInputDialog) and returns the value as a double. If the String does not contain a valid double value, then the program will terminate with an error.

(Note: Name the modified program as QuadraticCalculatorGUI.java and the output of the modified program looks like as follows. The expected output of the modified program is shown in Figure 5.)



(1) Input

Please enter the value of a:

1

(2) Input

Please enter the value of b:

-3.5

(3) Input

Please enter the value of c

2

(4) Message

The two solutions are
x1 = 2.7807764064044154
x2 = 0.7192235935955849

Figure 5: Expected output of the program (From step1 to step 4):

Marking Scheme

The marking of this exercise will be based on the following criteria.

Graded items	Weighting
1. Correctness of program (i.e. whether your code is implemented in a way according to the requirements as specified.)	50%
2. Presentation of the Java codes (i.e. whether the program is properly indented, how close you follow the common conventions as mentioned in class, etc.)	30%
3. Documentation (with reasonable amount of comments embedded in the code to enhance the readability)	20%
	100%

Be aware of plagiarism! DON'T copy the program file from your friends or classmates. If any identical copy is found, 5% of the coursework marks will be deducted for each of the file owner.

Program Submission Checklist

Before submitting your work, please check the following items to see you have done a decent job.

Items to be checked ☑ / ☒

1. Did I put my name, student ID and lab section at the beginning of all the source files as comment? □
2. Did I put reasonable amount of comments to describe my program? □
3. Are they all in .java extension and named according to the specification? □
4. Have I checked that all the submitted code are compliable and run without any errors? □
5. Did I zip my source files using Winzip / zip provided by Microsoft Windows? Also, did I check the zip file and see if it could be opened? ***(Only applicable if the work has to be submitted in zip format.)*** □
6. Did I submit my lab assignment to Canvas before the deadline? □