**Centennial College**

**COMP 228: Java Programming**

**LAB #2 – Java Methods**

**Student:** \_\_Waheed Alnoor\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Due Date: Week 5

Purpose: The purpose of this Lab assignment is to:

* Practice the use Java classes, Java methods, and other concepts taught.

References: Learning materials for week 3 and 4, textbook, and other references (if any)

This material provides the necessary information you need to complete the exercises.

Be sure to read the following general instructions carefully:

This lab should be completed individually by all the students.

YOU NEED TO SUBMIT THE FOLLOWING 2 DOCUMENTS & 1 COMMENT IN THE DROPBOX TITLED LAB2:

1. THE FIRST ONE IS A WORD DOCUMENT. USE THIS DOCUMENT AND ADD SCREEN SHOTS OF THE RUNNING STATE OF EACH EXERCISE (If there are more than 1 exercise). DO NOT DELETE THE QUESTIONS. THE SCREEN SHOTS SHOULD FOLLOW EACH QUESTION AND COVER ALL THE ASPECTS/FUNCTIONALITIES OF EACH EXERCISE. AFTER THE SCREEN SHOTS PLEASE COPY THE CODE FROM THE CODE WINDOW AND PASTE THE COMPLETE CODE. DO NOT GIVE ME SCREEN SHOTS OF THE CODE. DO NOT ZIP THIS FILE AND KEEP IT SEPARATE FROM YOUR ZIPPED PROGAM FILE.
2. SUBMIT ALSO ONE ZIPPED PROJECT FILE THAT CONTAINS ALL THE EXERISES SEPARATELY INTO THE SAME DROP BOX.
3. SUBMIT ALSO 1 COMMENT WITH YOUR GITHUB REPOSITORY LINK (MAKE SURE YOUR REPOSITORY VISIBILITY IS SET TO PUBLIC)

You must name your Eclipse/IntelliJ project according to the following rule:

* **YourFullName\_COMP228Labnumber**
  + Example: **JohSmith\_COMP228Lab2**
* Each exercise should be placed in a separate package named *exercise1*, *exercise2*, etc.
* Push all updates to your respective repository: **comp228406-FirstName-CurrentSemester**(i.e. comp228406-sohaib-fall2025)
  + Ensure your repository contains your project "**YourFullName\_COMP228Labnumber**"
* Submit your assignment in a **zip file** that is named according to the following rule:
  + **YourLastName\_COMP228Labnumber.zip**
  + Example: **JohSmith\_COMP228Lab2.zip**

Apply the naming conventions for variables, methods, classes, and packages:

- *variable names* start with a *lowercase* character

- *classes* start with an *uppercase* character

- **packages** use only *lowercase* characters

- *methods* start with a *lowercase* character

**Exercise 1:**

Write a Java application that simulates a test. The test contains **at least five** questions about first three lectures of this course. Each question should be a multiple-choice question with 4 options.

Design a **Test** class. Use programmer-defined methods to implement your solution. For example:

* create a method to simulate the questions – *simulateQuestion*
* create a method to check the answer – *checkAnswer*
* create a method to display a random message for the user – *generateMessage*
* create a method to interact with the user - *inputAnswer*

Display the questions using methods of ***JOptionPane*** class. Use a loop to show all the questions.

For each question:

* If the user finds the right answer, display a random congratulatory message (“Excellent!”,”Good!”,”Keep up the good work!”, or “Nice work!”).
* If the user responds incorrectly, display an appropriate message and the correct answer (“No. Please try again”, “Wrong. Try once more”, “Don't give up!”, “No. Keep trying..”).
* Use random-number generation to choose a number from 1 to 4 that will be used to select an appropriate response to each answer.
* Use a switch statement to issue the responses, as in the following code:

switch ( randomObject.nextInt( 4 ) )

{

case 0:

return( "Very good!" );

……

}

At the end of the test display the number of correct and incorrect answers, and the percentage of the correct answers.

Your main class will simply create a Test object and start the test by calling **inputAnswer** method.

(5 marks)

**Exercise 2:**

Design a Lotto class with one array instance variable to hold three random integer values (from 1 to 9). Include a constructor that randomly populates the array for a lotto object. Also, include a method in the class to return the array.

Use this class to simulate a simple lotto game in which the user chooses a number between 3 and 27. *The user runs the lotto up to 5 times and each time the sum of lotto numbers is calculated*. *If the number chosen by the user matches the sum*, the user wins and the game ends. *If the number does not match the sum within five rolls*, the computer wins.

Use methods of JOptionPane class to interact with the user.

(3 marks)

**Exercise 3:**

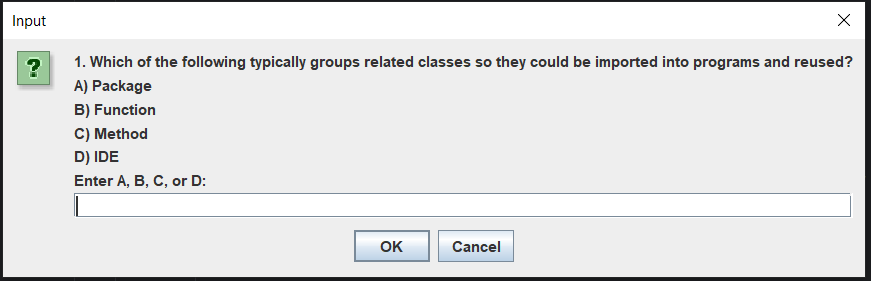
Write a Java class that implements a set of three overloaded static methods. The methods should have different set of parameters and perform similar functionalities. Call the methods within main method and display the results.

(2 marks)

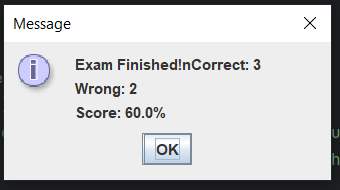
**Evaluation:**

|  |  |
| --- | --- |
| **Functionality** |  |
| Correct implementation of classes (instance variable declarations, constructors, getter and setter methods, etc.) | 40% |
| Correct implementation of driver classes (declaring and creating objects, calling their methods, interacting with user, displaying results) | 40% |
| Comments, correct naming of variables, methods, classes, etc. | 5% |
| **Friendly input/output** | 15% |
| **Total** | 100% |

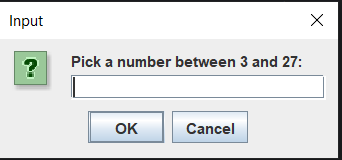
**Exercise 1 output**



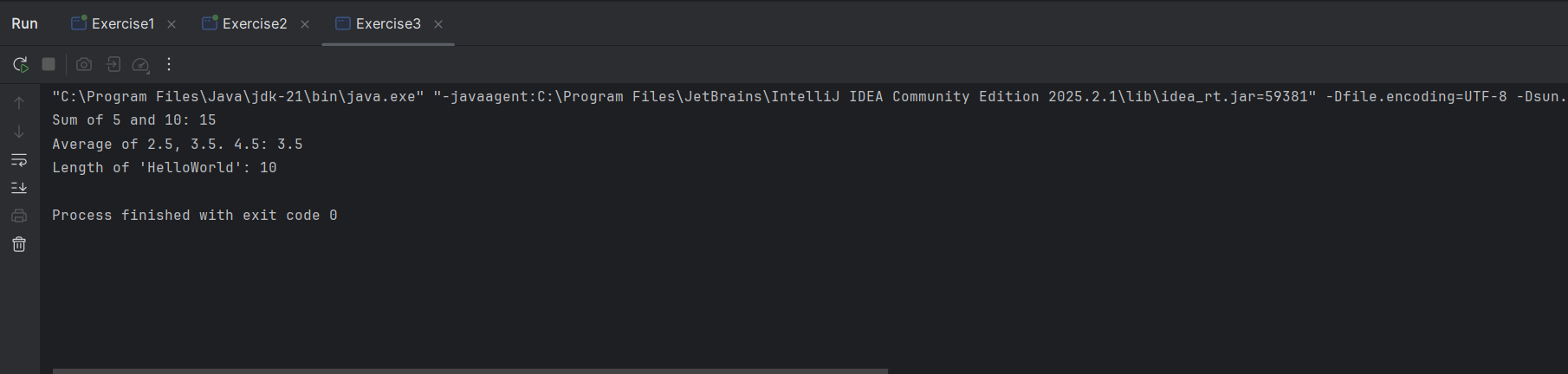
**Exercise 1 output**



**Exercise 2 output**



**Exercise 3 output**



**Code Exercise 1**

package exercise1;  
  
import javax.swing.JOptionPane;  
import java.util.Random;  
  
// the Main class  
  
public class Exercise1 {  
 public static void main(String[] args) {  
 Test exam = new Test();  
 exam.startExam();  
 }  
}  
// the questions  
class Test {  
 private String[] examQuestions = {  
 "1. Which of the following typically groups related classes so they could be imported into programs and reused?\n" +  
 "A) Package\nB) Function\nC) Method\nD) IDE",  
  
 "2. Method arguments may be \_\_\_\_\_\_\_\_\_ . \n" +  
 "A) constants, variables, or expressions.\nB) only variables\nC) only constants\nD) only strings",  
  
 "3. Class variables must be declared as \_\_\_\_\_\_ .\n" +  
 "A) final\nB) static\nC) const\nD) var",  
  
 "4. if a local variable or parameter in a method has the same name as a field of the class, the field is hidden until the terminates execution. This phenomenon is termed as \_\_\_\_\_ .\n" +  
 "A) shadowing\nB) buffering\nC) shadowcasting\nD) clouding",  
  
 "5. In which year did Sun Microsystems fund an internal corporate research project led by James Gosling, which resulted in a C++ -based object-oriented programming language that Sun called Java?\n" +  
 "A) 1999\nB) 1998\nC) 1991\nD) 1990"  
  
 };  
  
 // the correct answers  
  
 private char[] correctAnswers = {'A', 'A','B','A','C'};  
  
 private int rightCount = 0;  
 private int wrongCount = 0;  
 private Random random = new Random();  
  
 // the exam  
 public void startExam() {  
 for (int i = 0; i < examQuestions.length; i++) {  
 String input = JOptionPane.*showInputDialog*(  
 examQuestions[i] + "\nEnter A, B, C, or D:");  
  
 if (input == null || input.isEmpty()) {  
 JOptionPane.*showMessageDialog*(null, "No answer given. Skipping question.");  
 wrongCount++;  
 continue;  
 }  
  
 char ans = Character.*toUpperCase*(input.charAt(0));  
  
 if (isAnswerCorrect(i, ans)) {  
 rightCount++;  
 JOptionPane.*showMessageDialog*(null, getMessage(true));  
 } else {  
 wrongCount++;  
 JOptionPane.*showMessageDialog*(null,  
 getMessage(false) + "Correct answer:" + correctAnswers[i]);  
 }  
 }  
  
 double percentage = (rightCount \* 100.0) / examQuestions.length;  
  
 JOptionPane.*showMessageDialog*(null,  
 "Exam Finished!nCorrect: " + rightCount +  
 "\nWrong: " + wrongCount +  
 "\nScore: " + percentage + "%");  
 }  
  
 // check if its correct  
  
 private boolean isAnswerCorrect(int qIndex, char ans) {  
 return correctAnswers[qIndex] == ans;  
 }  
  
 // feedback message  
  
 private String getMessage(boolean isCorrect) {  
 switch (random.nextInt(4)) {  
 case 0: return isCorrect ? "Great job!" : "No,try again.";  
 case 1: return isCorrect ? "Well done!" : "; Oops, not correct.";  
 case 2: return isCorrect ? "Keep it up!" : " Dont give up !";  
 default: return isCorrect ? "Nice work!" : " Wrong this time.";  
 }  
 }  
}

**Code Exercise 2**

package exercise2;  
  
import javax.swing.JOptionPane;  
import java.util.Random;  
  
// generate 3 random number  
  
class Lotto {  
 private int[] numbers = new int[3];  
 private Random rand = new Random();  
  
 public Lotto() {  
 for (int i = 0; i < numbers.length; i++) {  
 numbers[i] = rand.nextInt(9) + 1;  
 }  
 }  
  
 // return the numbers  
 public int[] getNumbers() {  
 return numbers;  
 }  
  
 public int getSum() {  
 int sum = 0;  
 for (int n : numbers) {  
 sum += n;  
 }  
 return sum;  
 }  
}  
 public class Exercise2 {  
 public static void main(String[] args) {  
 // choosing number  
 int userChoice = Integer.*parseInt*(  
 JOptionPane.*showInputDialog*("Pick a number between 3 and 27:")  
 );  
  
 boolean userWon = false;  
  
 // 5 tries  
 for (int i = 1; i <= 5; i ++) {  
 Lotto lotto = new Lotto();  
 int sum = lotto.getSum();  
  
 // the result  
  
 JOptionPane.*showMessageDialog*(null,  
 "Roll " + i + ": " +  
 lotto.getNumbers()[0] + "+" +  
 lotto.getNumbers()[1] + "+" +  
 lotto.getNumbers()[2] + "+" + sum);  
 // check if it right  
  
 if (sum == userChoice) {  
 JOptionPane.*showMessageDialog*(null, " You Win Its matched your number.");  
 userWon = true;  
 break;  
 }  
 }  
 // if its not match  
  
 if (!userWon) {  
 JOptionPane.*showMessageDialog*(null, "Computer wins No match within 5 rolls.");  
 }  
 }  
}

**Code Exercise 3**

package exercise3;  
  
public class Exercise3 {  
  
 public static int calculate(int a, int b) {  
 return a + b;  
 }  
  
 public static double calculate(double a, double b, double c) {  
 return (a + b + c) / 3;  
 }  
  
 public static int calculate(String text) {  
 return text.length();  
 }  
  
 public static void main(String[] args) {  
  
 System.*out*.println("Sum of 5 and 10: " + *calculate*(5, 10));  
 System.*out*.println("Average of 2.5, 3.5. 4.5: " + *calculate*(2.5, 3.5, 4.5));  
 System.*out*.println("Length of 'HelloWorld': " + *calculate*("HelloWorld"));  
 }  
}