

## **GUI Programming 2019-2020 – Year 2**

### **Labwork 2 - Java Revision 2 (Methods):**

**(Worth 5% - or 50 points out of 500 points for labwork this semester)**

#### **IMPORTANT NOTES:**

- **NO COPYING PERMITTED AND ZERO MARKS WILL APPLY TO COPIED WORK. FURTHER ACTION MAY BE TAKEN AGAINST STUDENTS THAT HAVE BEEN FOUND TO COPY WORK.**
- **ASSESSMENT WILL INVOLVE ONE-TO-ONE QUESTIONS ABOUT YOUR SUBMITTED WORK. USE COMMENTS IN YOUR CODE TO ENSURE YOU DON'T FORGET WHY YOU WROTE CODE YOU MAY LATER BE ASKED ABOUT.**
- **ALL WORK MUST BE SUBMITTED TO MOODLE BY DATES SPECIFIED (2 LABS SUBMISSIONS OF FIVE LABS THROUGHOUT THE SEMESTER).**
- **MANY OF THE TASKS ASSIGNED BELOW CAN BE COMPLEX AND\OR THE DESCRIPTIONS MAY REQUIRE FURTHER CLARIFICATIONS. PLEASE USE THE AVAILABLE LAB TIMES TO ASK FOR CLARIFICATIONS AND ADVICE\HINTS ON THE TASKS BELOW.**

### Part 1 – Basic Method – no return (a void method) (5 points)

Create a Java program called **Lab2Part1** that defines a method called **checkAge** to input an integer representing your age. The method must output a message to the screen (use System.out) if the age passed is “less than”, “equal” or “greater than” your actual age (you could set your age as a final static variable in the program so that when your age changes the program will still work once the final static variable is changed).

Required activities and marking guideline:

- Create the method with correct name (2 points)
- Pass the integer as a parameter to the method (1 point)
- Output the correct message based on your actual age (2 points)

### Part 2 – Basic Method – return boolean (non-void method) (7 points)

Create a Java program called **Lab2Part2** that creates a method called **canVote** that inputs your age and returns **true** or **false** to the calling method depending on whether you can vote or not ( $\geq 18$ ). You must test your method by calling it from the main and passing various ages to it, *if(canVote(17))* etc.

Required activities and marking guideline:

- Create the method with correct name (2 points)
- Pass the integer as a parameter to the method (1 point)
- Return the boolean result using **return** statement (2 points)
- Test the method at least twice by calling it in main (2 points)

### Part 3 – Basic Method – passing parameter return *int*(non-void) (10 points)

Create a Java program called **Lab2Part3** that contains a method called **calculateMyNameCharacterValue**. This method should **receive a character array** containing your full name and return the total numeric value of your name as one integer, e.g., the character value of the name ‘A’, ‘B’, ‘E’ is  $65 + 66 + 69$ . Passing the array {‘A’,‘B’,‘E’} would return 200 to the calling code.

Required activities and marking guideline:

- Create the name array (2 points)
- Create the method with correct name used (2 points)
- Include correct parameter for method (2 points)
- Include correct return type for the method (1 point)
- Do the calculation of the character values and return (2 points)
- Test and output the result with your full name (1 point)

#### Part 4 – Method with multiple parameters (10 points)

Create a Java program called **Lab2Part4** that contains a method called **outputInReverse**. The method must input THREE strings and then output the three strings in reverse with a space between them, e.g., **outputInReverse**("The", "Black", "Cat") will output "Cat Black The" as one string.

Required activities and marking guideline:

- Create the method with the correct name (2 points)
- Define all three parameters (3 points)
- Write code to set new strings in reverse order and output (3 points)
- Test the method by calling it at least twice with strings (2 points)

#### Part 5 – Creating custom methods (More Advanced) (18 points)

Create a Java program called **Lab2Part5** that implements a toll bridge charging system for different vehicle types. The system must have TWO METHODS as follows:

- Write a method called **getVehicleCharge()** that will input the type of vehicle (as a String) and return the amount to charge for that vehicle as a double
- Write a method called **outputTotalTolls()** that will input an array of vehicles, e.g., {"Car", "Car", "Motorbike" ... etc., and will output the total charges the toll bridge has collected for vehicle array using System.out

The **outputTotalTolls()** method will input the array of vehicle and then must call the **getVehicleCharge()** method each time in sequence (which will return the correct toll to charge) until it adds up the total for all vehicles and outputs it as a double to System.out. The charges for the toll are as follows (You can make your own version of charges if you want but you must have at least five different types of vehicle and five different charges):

"Motorbike"	10c
"Car"	20c
"Van"	30c
"Bus"	50c
"Truck"	100c

Required activities and marking guideline:

- Create the getVehicleCharge method to return charge per vehicle (6 points)
- Create the outputTotalTolls method to return the total tolls (6 points)
- Call the getVehicleCharge method from outputTotalTolls method (2 points)
- Create a test array with at least 10 vehicle types listed (2 points)
- Test the outputTotalTolls method using the vehicle array (2 points)