| SHADED logo | INSTITUTE OF TECHNOLOGY TRALEE  WINTER EXAMINATIONS AY 2012-2013  **Introduction to Programming**  **PROG61001**  **CRN 43824** |
| --- | --- |

**External Examiner**:

**Internal Examiner**: Mr. J. Brosnan

**Duration**: 2 Hours

**Instructions to Candidates:** You may attempt *any* 3 questions. All questions carry equal marks. Show all workings clearly as you may lose marks otherwise.

**Q1.**

**(a)**

Write a Java program that first of all contains a single line comment stating the file name, which you can take to be **House.java** and a brief multi-line comment, to explain its operation. The program should then **create four variables** to store the following details for a particular house being advertised by an estate agency:

House Location: Oakpark Demense, Tralee, Co. Kerry

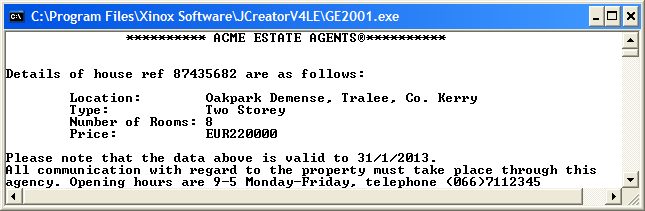
House Type: Two Storey

Number of Rooms: 8

House Price: €220,000

Note that you should use the most appropriate type of variable in each case above i.e. string or numeric (use numeric variables wherever possible) and give the variables the most meaningful names you can.

Once the details above have been set, your program should produce **exactly** the output indicated on the screenshot below, including all formatting, tabs and blank lines. Note that the ASCII code for the registration symbol (®) used in the banner is 169. Your program should use a **single println() call** to display the information shown. Naturally, the variables referred to earlier should be used in the println() statement in order to display the desired information.



The \* are in 15 places from the left edge.

Location:, Type: etc are in 1 tab

Oakpark, Two Storey etc. are in a further 17 places

**(21 marks)**

**(b)**

Write a Java program that determines the period of a pendulum, which is the time taken for it to complete one full cycle in its motion.

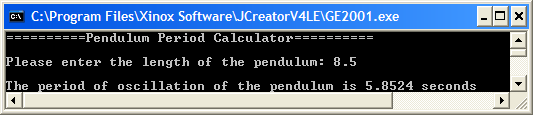
The period, T, of the pendulum is given by the formula

where l is the length of the pendulum, g is the acceleration due to gravity, which on Earth is 9.8 m/s/s, and π is the number pi whose value can be set to 3.142 for the purposes of this exam. You should **create constants** for g and π in your program and set them accordingly.

The program should request the user to supply a value for the length of the pendulum and then use the formula above to calculate the corresponding period. You can use **Math.pow()**here in your calculation.

The period should then be displayed correct to **4 decimal places**.

Your program should execute exactly as indicated in the sample run below:



**(12.33 marks)**

**Q2.**

**(a)**

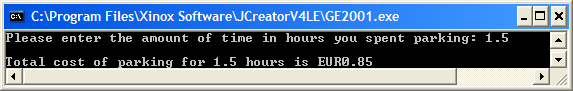
A person entering a parking lot receives a ticket from a machine and when they leave the parking lot the machine takes the ticket, determines the amount of time spent parking and calculates the amount the customer must pay. It uses the following table of values in its calculations:

|  |  |
| --- | --- |
| **Time Spent Parking** | **Cost** |
| Up to and including 1 hour | 60c per hour |
| Next two hours | 50c per hour |
| Anything more | 40c per hour |

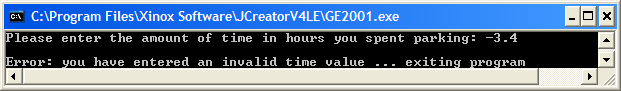
So, for example, if a person spends 1.5 hours in the parking lot, they must pay 60c for the 1st hour and 25c for the remaining half hour for a total of 85c.

Your program should **use only if structures** for decision-making purposes and execute as indicated in the sample runs below, displaying the total cost correct to 2 decimal places. There should also be some **fool-proofing** in your program so that if a negative number of hours is entered, an appropriate error message is displayed and the program immediately terminates.

**Here a valid time is supplied**



**Here an invalid time is supplied**

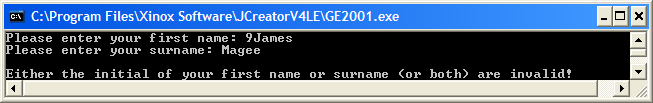


**(16.33 marks)**

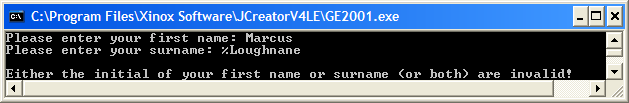
Write a Just BASIC program that asks the user to enter their first name and then their surname. Once these have been entered, both should be converted to lowercase. Next, the first character (initial) of each should be extracted and checked to see whether or not it is a valid letter of the lowercase alphabet. If either initial is invalid then the program should give an appropriate error message and immediately terminate. If, however, both initials are valid then the program should simply display the initials of the user to the screen (in lowercase is fine here).

The program should run according to the following sample screenshots:

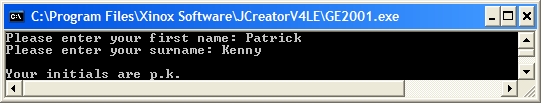
**In this run the first character of the first name is invalid as it begins with a number**



**In this run the first character of the surname is invalid as it begins with a % sign**



**In this run both initials are valid**



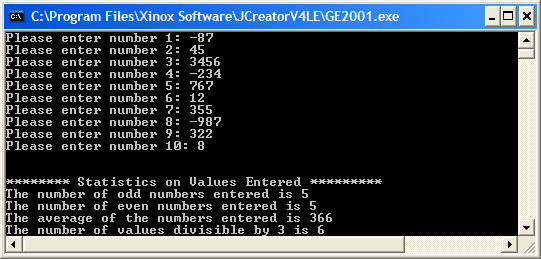
**(17 marks)**

**Q3.**

Write a Java program that uses a **while** loop to read in exactly 10 whole numbers. Once the numbers have been entered, the program should determine and display

* The number of values entered that were odd numbers
* The number of values entered that were even numbers
* The average of the values entered, to the nearest whole number
* The number of values entered that were exactly divisible by 3

Your program should run as indicated in the following sample screenshot:



**(16.33 marks)**

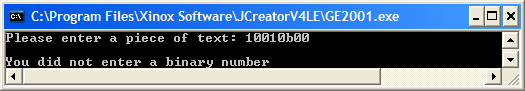
**(b)**

All valid binary numbers should only contain the characters 0 and 1. Write a Java program that asks the user to enter a piece of text and validates it to see whether it constitutes a binary number. The program should then display whether the value entered was a binary number or not.

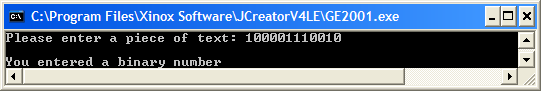
You should use a **do-while** loop when coding your solution here. You should find the **charAt**() method useful when coding your solution.

Your program should run as indicated in the sample screenshots below:

**In this run the text entered was not a binary number**



**In this run the text entered was a binary number**



**(17 marks)**

**Q4.**

A currency converter application must be written that provides the following options:

1. Convert Euro to British Pounds
2. Convert British Pounds to Euro
3. Quit

The application will repeat continuously until the user decides to quit, by selecting option 3 above. You should use a **for** loop for this process. When the user has been asked for their choice from 1-3 above, this choice value must firstly be validated to ensure it is a whole number within the range 1-3 inclusive. If it fails this tests, the user will be continuously prompted to re-enter their choice until a valid one is provided (you may use a **while** loop for this validation process).

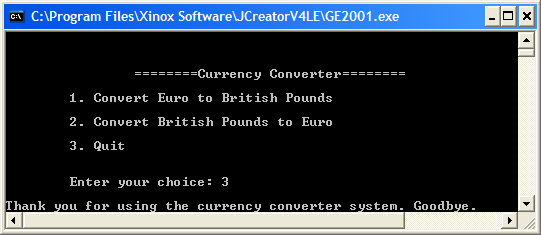
When a valid choice has been supplied, the appropriate course of action will then occur – a **switch** structure should be used to make the decision in this case. In cases 1 or 2, the user will be prompted for either a quantity of Euro or British Pounds respectively. The Euro quantity entered must be validated to ensure it is a valid positive floating-point value greater than or equal to zero (again, you can use a **while** loop here if you wish). As long as the user enters an invalid Euro amount, they will be continuously prompted to re-enter a valid amount. You do not need to validate the British Pounds value entered at all for this exam.

Finally, once a valid quantity in Euro/Pounds is supplied, the conversion will be performed and all results will be displayed to **2 decimal places**.

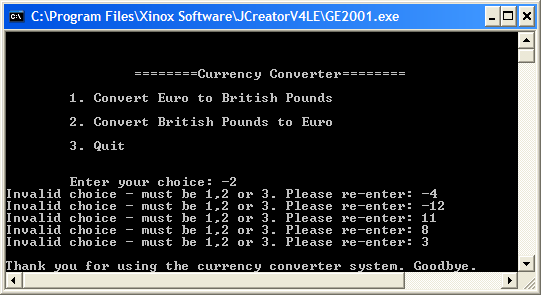
Note that **EUR1 = 0.919161 British Pounds** for the purposes of this question.

Your program should run **exactly** as indicated in the following sample screenshots, including banners, blank lines, tabs and cleared screens. You can take it the menu is in 2 tabs from the left edge of the window.

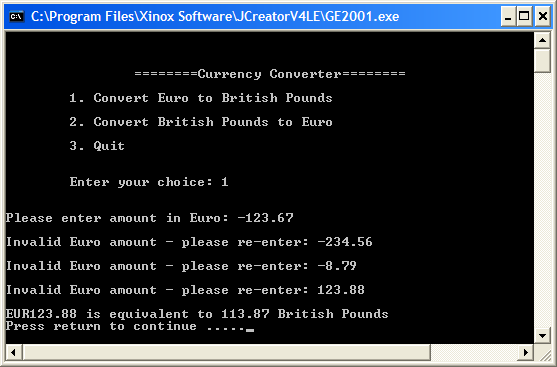
**Run 1: In this case the application launches and the user immediately selects the quit option - a farewell message is issued and the program terminates immediately**



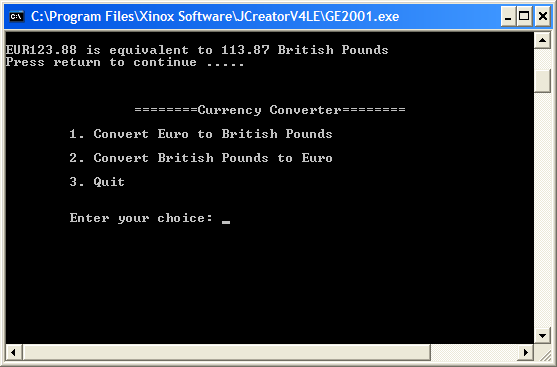
**Run 2: Here the application launches and the user enters a number of invalid choice values before eventually entering a valid one (option 3 in this case).**



**Run 3: Here the user selects option 1. They are then prompted for an amount in Euro and the program halts until the user supplies a value. In this case some invalid values have been supplied and all are rejected until a valid one is given. The conversion is then performed and results displayed to 2 decimal places. The program then halts until the user hits return**



**and as soon as the user hits return, the main (for) loop repeats, presenting the user once more with the original menu of options**



**(33.33 marks)**