****

**Assessment Brief**

|  |  |
| --- | --- |
| **Module Title:** | **Programming and Design Principles** |
| **Module Code**: | **5N2927** |
| **Assessment Technique**: | **Skills Demonstration** |
| **Weighting**:  **Issue Date** | **30%**  **15/11/2023** |
| **Deadline:** | **13th December 2023** |

**Guidelines:**

You will devise a program according to the attached brief which will use each of the following:

* the conditional statement (selection structure)
* a loop construct (iteration structure)
* input and output statements
* a selection of math, relational and Boolean operators
* different data types

**You have 10 hours to complete this project**

I confirm, that the attached is all my own work and does not include any work completed by anyone other than myself. I have completed the work in accordance with instructions given and within the time limits set by my center.

**Learner Signature:**  \_\_\_\_\_\_\_\_\_\_\_\_ **Date**: \_\_\_\_\_\_\_

**Teacher Signature:**  **Date**:

Assessment Brief

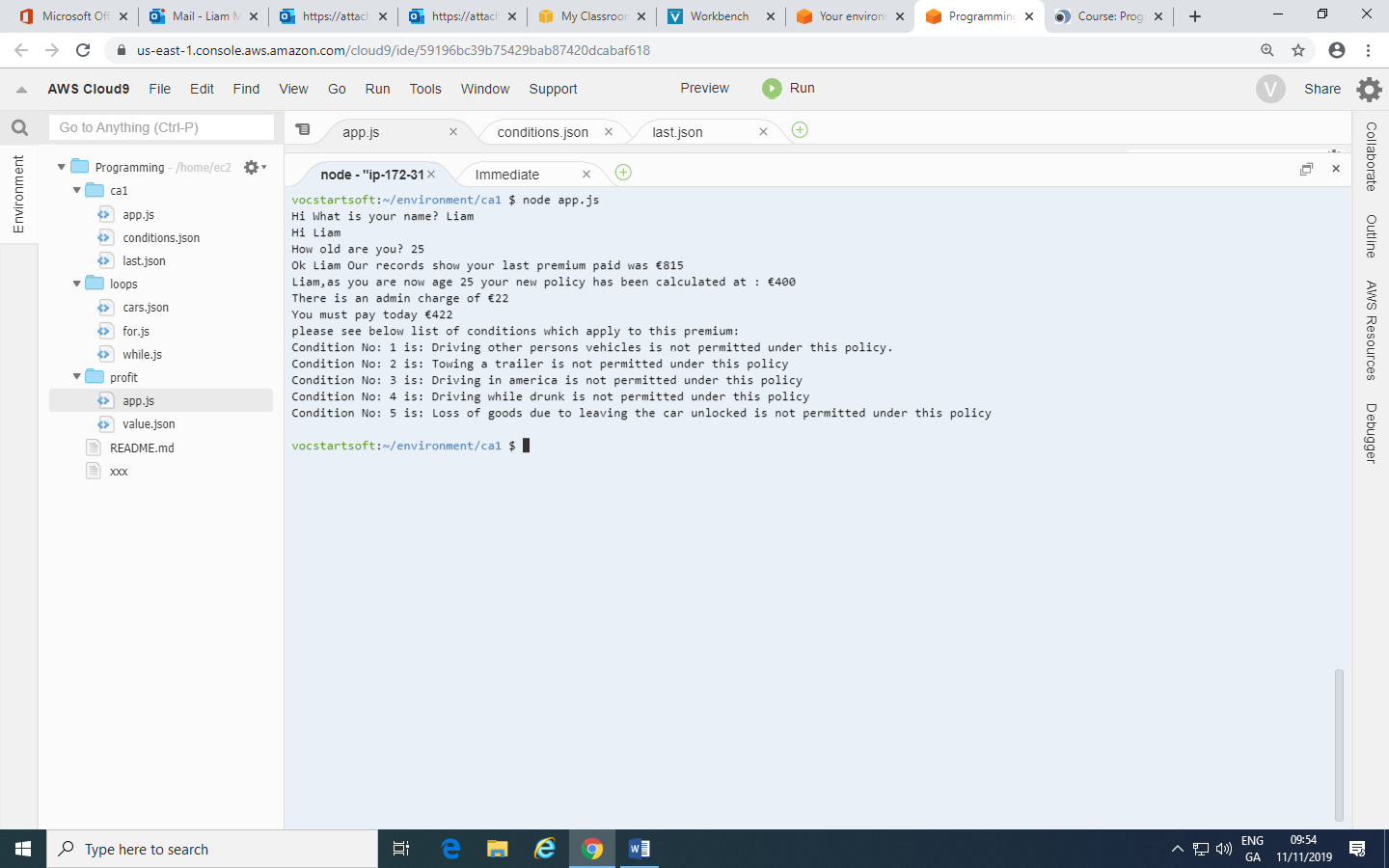
**This assignment is worth 30% of the course**

Write a program that when it is run it will do the following:

Present the user with some options as follows

1. Ask the user for an input for their name.
2. Output the users name in the terminal along with the value they paid for the last policy they renewed (from an external file).
3. Take a second input and ask the user their age.
4. Output the name and age in a sentence
5. Take the users age if they are less than 20 or are actually 20 tell them they are two young to get a policy and inform them how many years they must wait to get a policy. If they are between 21 and 35 multiply their age by 16 to get the premium price and if they are older than 35 tell them the policy will cost €700
6. Output that the user’s insurance premium which must include an addition of a €22 administration charge (This charge must be stored as a variable and not hard coded into the application)
7. Print a list of conditions from an external file that the policy contains (min 5 e.g. Fully Comprehensive, 2. Cannot drive other person’s vehicles etc.)

The answer should look something like this

AN INTRODUCTION TO PYT GCSE COMPUTING

***Familiarize yourself with the marking scheme before you begin development of the program.***

**Development process**

1. Create a **flowchart** on paper or in draw.io or a similar application
2. Create and document a suitable **data dictionary** (a collection of descriptions of the data objects or items in a data model for the benefit of programmers and others who need to refer to them).
3. Create a **test plan** with examples of the data that you will test and the list of expected outcomes. Make sure to test invalid options.
4. Create your **solution** in JavaScript including sufficient comments
5. Remember to **record examples of errors** you found and how you fixed them
6. Run your **test scenarios** and take screenshots
7. You may add elements to **improve** the program- indicate why you added something and what the benefit was

Checklist

* Flowchart
* Data Dictionary
* Test Plan
* Fully Commented JavaScript Code
* Problems Encountered and how they can be solved
* Testing Evidence
* Details of an enhancements to original brief

|  |  |  |
| --- | --- | --- |
| **Assessment Criteria** | **Maximum Mark** | **Learner Mark** |
| Skills Demonstration 1   * Algorithm   + Pseudocode or flowchart detailing all tasks to be completed.   + Complete data dictionary compiled. * Accurate programming   + Program compiles.   + Appropriate data types chosen for variables.   + Correct use of input statements with suitable input prompts.   + Correct use of output statements with output appropriately labelled.   + Correct use of mathematical, relational and Boolean operators.   + Correct use of selection structures.   + Correct use of iteration structures. * Appropriate testing   + Suitable test data compiled.   + All computation on test data shown.   + Correct results shown for each piece of test data.   + Screen shots included that show results of compiled test data used on coded solution. * Accepted industry standards for coding   + Logical sequence to program.   + Code suitably commented.   + Indenting conforms to industry standard.   + Clear and consistent input prompts given to user.   + Clear and consistent output from the program, suitably displayed. | 5  15  5  5 |  |
| **Total Mark** | **30** |  |