# PROG 1700 - Assignment 2

# DECISIONS AND FUNCTIONS

Assignment Value: 10% of overall course mark.

Due Date: **See due date designated on the assignment dropbox on Brightspace.**

Late submissions will receive the standard late submission penalty as stated in the course outline. (5% overall deduction per day late, until 60%, and 0% after assignment handed back to the class.)

#### Assignment Instructions:

Create console applications (.py files) in which you’ll code the answer for each of the following problems. **You must create a new .py file for each question in this assignment.**

#### Submissions:

#### You will submit your work for this assignment via GitHub. While you will have frequent commits/pushes of your assignment code to GitHub as your work on it, the instructor needs to know which version to mark and when it was committed. So, when you have completed all assignment work, put a “Ready for Marking” comment on the last code committed into GitHub. Then submit a simple text document to the Brightspace Dropbox that contains the git Commit ID string (e.g. “b180b37”) that identifies that commit. It is this Dropbox submission that will be used to determine late penalties, so make sure to do so prior to the assignment deadline.

#### Once you have committed the code, make sure to visit the repository page on GitHub’s website to verify that the final version has been pushed to GitHub as that is where the instructor will go to get your code.

#### Evaluation:

To insure the greatest chance of success on this assignment, be sure to check the marking rubric contained at the end of this document or in Brightspace. The rubric contains the criteria your instructor will be assessing when marking your assignment.

Program 1 – Landscape Calculator

A landscaping company needs a program that computes the price of landscaping for a new housing development. Work orders are based on: address, plot length and width in feet, type of grass (“fescue”, “bentgrass” or “campus”), and number of trees. The price is computed as follows:

* There is a base labour charge of $1000.
* If the surface (length \* width) is over 5000 square feet, add $500.
* The cost is calculated per square foot. If the grass is “fescue” the cost is $0.05; for “bentgrass” it is $0.02; “campus” is $0.01.
* Each tree requested has a $100 charge.

First, create a flowchart that clearly shows all the paths of execution that will exist within your designed solution to this problem. Write a console application that will input the address, property length and width, type of grass and number of trees, and then output the corresponding price.

Your solution must contain examples demonstrating your understanding of appropriate use of functions and core assignment concepts (decision structures).

EXAMPLES AND TESTING

In the section below you will be presented with at least one screenshot of a successful execution of a sample solution to the program, which should help demonstrate how your input/output on the program should work. In addition to the sample values used in the screenshot(s), additional testing values are given in a chart along with the output values that they should produce. You can expect your instructor to grade your assignment by using all of these listed input values at the very least, but keep in mind that additional values may also be used as well. In other words,**you should thoroughly test your code before submitting!**

**Sample Output** - Make sure your program can output data *exactly* as shown below.

|  |
| --- |
| **Enter House Number: 439**  **Enter property depth (feet): 100**  **Enter property width (feet): 50**  **Enter type of grass (fescue, bentgrass, campus): fescue**  **Enter the number of trees: 2**  **Total cost for house 439 is: $1450.00** |

**Other testing values**;

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **House Number** | **Depth** | **Width** | **Type** | **No. of Trees** | **Total Cost** |
| **55** | 150 | 40 | bentgrass | 0 | $1620.00 |
| **10** | 75 | 51 | campus | 1 | $1138.25 |

Program 2 – Erewhon Mobile Data Plans

Erewhon Mobile charges cellular customers a rate based on the total amount of data used by a customer in the billing period. For simplicity, customers are charge based on which range their total data usage falls within. Note, it is not a cumulative charge; your program will figure out which single range the usage falls into, then calculate the cost based on that range’s cost.

|  |  |
| --- | --- |
| **Total Data Usage** | **Rate of Charge** |
| **Up to and including 200Mb** | $20.00 flat rate |
| **Over 200Mb and up to and including 500Mb** | $0.105 per Mb |
| **Over 500Mb and up to and including 1Gb** | $0.110 per Mb |
| **Over 1Gb** | $118.00 flat rate |

Then, write a console application to input the customer’s usage in mega/gigabytes and output the corresponding rate and charge.

Your solution must contain examples demonstrating your understanding of appropriate use of functions and core assignment concepts (decision structures).

EXAMPLES AND TESTING

In the section below you will be presented with at least one screenshot of a successful execution of a sample solution to the program, which should help demonstrate how your input/output on the program should work. In addition to the sample values used in the screenshot(s), additional testing values are given in a chart along with the output values that they should produce. You can expect your instructor to grade your assignment by using all of these listed input values at the very least, but keep in mind that additional values may also be used as well. In other words,**you should thoroughly test your code before submitting!**

**Sample Output** - Make sure your program can output data *exactly* as shown below.

|  |
| --- |
| **Enter data usage (Mb): 250**  **Total charge is $26.25** |

**Other testing values:**

|  |  |
| --- | --- |
| **Data Usage** | **Total Charge** |
| **175** | $20.00 |
| **436** | $45.78 |
| **2035** | $118.00 |

Program 3 – Auto Insurance

Write a program that computes monthly insurance according to the following schedule:

|  |  |  |
| --- | --- | --- |
| **If you are ‘Male’ and your age is** | **But less than** | **the price of the vehicle multiplied by** |
| 15 or greater | 25 | 25% / 12 |
| 25 | 40 | 17% / 12 |
| 40 | 70 | 10% / 12 |
| **If you are ‘Female’ and your age is** | **But less than** | **the price of the vehicle multiplied by** |
| 15 or greater | 25 | 20% / 12 |
| 25 | 40 | 15% / 12 |
| 40 | 70 | 10% / 12 |

First, create a flowchart that clearly shows all of the paths of execution that will exist within your designed solution to this problem. Then, write a console application that will input the sex and price of vehicle, and then output the corresponding monthly insurance amount.

Your solution must contain examples demonstrating your understanding of appropriate use of functions and core assignment concepts (decision structures).

EXAMPLES AND TESTING

In the section below you will be presented with at least one screenshot of a successful execution of a sample solution to the program, which should help demonstrate how your input/output on the program should work. In addition to the sample values used in the screenshot(s), additional testing values are given in a chart along with the output values that they should produce. You can expect your instructor to grade your assignment by using all of these listed input values at the very least, but keep in mind that additional values may also be used as well. In other words,**you should thoroughly test your code before submitting!**

**Sample Output** - Make sure your program can output data *exactly* as shown below.

|  |
| --- |
| Are you 'Male' or 'Female': Male  Enter your age: 27  Enter the purchase price of the vehicle: 6500  Your monthly insurance will be $92.08 |

**Other testing values**;

|  |  |  |  |
| --- | --- | --- | --- |
| **Sex** | **Age** | **Vehicle Price** | **Monthly Insurance** |
| **Male** | 23 | 500 | $10.42 |
| **Male** | 34 | 2500 | $35.42 |
| **Female** | 26 | 999 | $12.49 |
| **Female** | 46 | 3200 | $26.67 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Program 1 – Landscape Calculator** | | |  |  |  |  |
| **Criteria** | **Insufficient (0 pts)** | **Needs Development**  **(1-2 pts)** | **Sufficient (3-4 pts)** | **Excellent (5 pts)** | **Mark** | **X** |
| **Flowchart** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | A comprehensive effort was made to plan out the program using a flowchart that demonstrates the necessary paths through the program based on all potential decision points. |  |  |
| **Input / Output** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | The five user input values can be successfully captured through descriptive prompts. The single output line is well-formatted and contains all expected information. Output amounts displayed using proper currency formatting (e.g. preceded by a $ symbol, two decimal places) |  |  |
|  |
| **Variables & Data Types** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Solution displays strong understanding of variable usage. All data is stored using correct data types and cast to other data types when required. |  |  |
| **Grass Type** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Grass Type always determined from user input and program can handle upper and lower case entries, or any variation thereof. |  |  |
| **Total Cost** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Total cost amount is correctly calculated and output to the console. |  | 2 |
| **Use of Functions** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Solution contains at least one example demonstrating strong understanding of appropriate function use, including use of parameters and return values, and with no global variable use. (Using or adapting the standard main() function does not count.) |  |  |
| **Comments & Best Coding Practices**  (At least 60% of the functional requirements must be complete) | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Organizational or explanatory comments are used extensively, most are meaningful and easily understood. A consistent naming convention was used for most of the program and deviated very little. Source code was clean, consistently well-formatted and easy to read |  | 2 |
|  | | | | **Total**: |  | **/45** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
| **Program 2 – Erewhon Mobile Data Plans** | | |  |  |  |  |
| **Criteria** | **Insufficient (0 pts)** | **Needs Development**  **(1-2 pts)** | **Sufficient (3-4 pts)** | **Excellent (5 pts)** | **Marks** | **X** |
| **Input / Output** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | All of:   * The single user input value can be successfully captured through a descriptive prompt. * The single output line is well-formatted and contains all expected information * Output amount displays using proper currency formatting (e.g. preceded by a $ symbol, two decimal places) |  |  |
| **Variables & Data Types** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Solution displays strong understanding of variable usage. All data is stored using correct data types and cast to other data types when required. |  |  |
| **Total Charge** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Total charge amount is correctly calculated based on variable values. |  | 2 |
| **Use of Functions** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Solution contains at least one example demonstrating strong understanding of appropriate function use, including use of parameters and return values, and with no global variable use. (Using or adapting the standard main() function does not count.) |  |  |
| **Comments & Best Coding Practices**  (At least 60% of the functional requirements must be complete) | Little to no effort was made, or contains too many errors / omissions. | A good effort was made, but at least one error or omission exists. | A good effort was made, but at least one error or omission exists. | Organizational or explanatory comments are used extensively, most are meaningful and easily understood. A consistent naming convention was used for most of the program and deviated very little. Source code was clean, consistently well-formatted and easy to read. |  | 2 |
|  |  |  |  | **Total:** |  | **/40** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Program 3 – Auto Insurance** | |  |  |  |  |  |
| **Criteria** | **Insufficient (0 pts)** | **Needs Development**  **(1-2 pts)** | **Sufficient (3-4 pts)** | **Excellent (5 pts)** | **Marks** | **X** |
| **Flowchart** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | A comprehensive effort was made to plan out the program using a flowchart that demonstrates the necessary paths through the program based on all potential decision points. |  |  |
| **Input / Output** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Both of:  The three user input values can be successfully captured  through descriptive prompts  The single output line contains all expected information and formatted as shown |  |  |
| **Variables & Data Types** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Solution displays strong understanding of variable usage. All data is stored using correct data types and cast to other data types when required. |  |  |
| **Male or Female** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Program correctly decides/branches based on sex. |  |  |
| **Insurance Calculations** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Final insurance price is correctly calculated based on user-entered values. |  | 3 |
| **Use of Functions** | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Solution contains at least one example demonstrating strong understanding of appropriate function use, including use of parameters and return values, and with no global variable use. (Using or adapting the standard main() function does not count.) |  |  |
| **Comments & Best Coding Practices**  (At least 60% of the functional requirements must be complete) | Little to no effort was made, or contains too many errors / omissions. | A reasonable effort was made, but there are multiple areas for improvement. | A good effort was made, but at least one error or omission exists. | Organizational or explanatory comments are used extensively, most are meaningful and easily understood.  A consistent naming convention was used for most of the program and deviated very little. Source code was clean, consistently well-formatted and easy to read. |  | 2 |
|  |  |  |  | **Total:** |  | **/50** |