Lesson 5

Assignment 5A – working with the while and decision structure

# Objectives:

* Use the while structure to keep track of:
  + An accumulator
  + A counter
* Use the decision structure to control/manipulate the final output
* Output the results

# assignment Task Checklist

* It is expected you have read the required reading in this Lesson before starting the lab.
* Review Course Resources:
  + Rules of Engagement (ROE).
  + Guidance/Menu Guidance
* Create the required algorithm to meet customer requirements in Flowgorithm
* Ensure you have test cases reflected in the comments
* File naming Convention: “lastname-asgn5a.fprg”.
* Submit all required lab files to blackboard

# assignment Task Checklist

1. Open Flowgorithm and save the file with the required naming convention.
2. Review the customer requirements

***Customer Requirements:***  *The customer needs an application that creates a report telling how many months it will take to pay of a loan given a loan payment.*

*The application will accept a client’s loan amount and monthly payment amount. It will then output the customer’s loan balance each month until the loan is paid off. The report should not have any negative numbers (the final month should reflect a balance of $0).*

*Display the month and remaining balance for each month until paid off. In addition, the program should display what the original loan amount was.*

1. Review the example IPO and Hierarchy Chart “Additional Lab Materials” below.
2. Open Flowgorithm and save the file with the required naming convention.
   1. Enter the required program attributes for your program.
   2. Create the algorithm using Flowgorithm to meet the customer requirements
      * Structure your program using the Hierarchy Chart below.
      * Make sure you are using the correct datatype
      * Test using the IPO test cases and document the test case in your code as comments.
3. Additional Requirements:
   1. Create modules/functions that:
      * Display a welcoming message and a purpose statement
      * Processing for shipping costs
      * Displays an End of job message and your name
   2. Processing will require:
      * A counter for month
      * An accumulator for balance \*hint: initialize with the loan amount
      * Logic to ensure the last month reflects a 0 balance and not a negative number (see month 11 example in render).
      * A message with the Original Loan Amount before end of program.

# Additional Lab Materials:

### **Hierarchy Chart**

### **IPO**

| **Input** | **Processing** | **Output** |
| --- | --- | --- |
| Loan Amount  Planned monthly payment | Get loan amount  Get planned monthly payment  Assign counter  Assign original loan amount  Test for balance  Calculate new balance  Keep track of month  Test for 0 balance | Welcome message  Prompt: loan amount, and monthly payment  Result: month and balance for each month  Original Loan amount  End of Program message |
| **Storage/Memory Location**  The student will need to decide which names to use. Remember to follow the correct naming conventions. | The student will need to decide which names to use. Remember to follow the correct naming conventions |  |
| **Test Data** | T1: LA = 1001 Pymt = 100… 11 months till 0.  T2: LA =2079.50 Pymt = 231.20…9 months till 0  T3: LA = 4000 Pymt 250…16 months till 0 |  |

1. Expected Render/Output **Example**. Do not hardcode or use magic numbers. Your code should allow different loan amounts and payments:

|  |  |
| --- | --- |
| Graphic of expected render of code | Output Results:  Welcome to Know Your Loan Company  Enter the loan amount: 1001  Enter the payment amount: 100  Month 1 Balance is 901  Month 2 Balance is 801  Month 3 Balance is 701  Month 4 Balance is 601  Month 5 Balance is 501  Month 6 Balance is 401  Month 7 Balance is 301  Month 8 Balance is 201  Month 9 Balance is 101  Month 10 Balance is 1  Month 11 Balance is 0  Your Original Loan Amount was : 1001  Thank you for using this program |