**Understanding**:

***What is the assignment asking of us?***

For this first project, we are tasked with calculating the monthly payment on a loan.

***What techniques are new this week and required to complete the project? What are the techniques?***

* 3 primary activities of a program: Input, processing, output
* Comments: use of //, ignored by compiler, used to explain the program to a reader. For comments that span multiple lines, /\* marks the beginning and \*/ marks the end of the comment.
* Pre-processor directive: Reads the program and only executes on lines with ‘#’. A program that sets up the source code for the compiler.
* #include <iostream>: Causes the pre-processor to include contents from another file. The name of the file is inside the brackets.
* Name space std: A variable that declares that the program will be accessing entities whose names are part of the namespace called std.
* Function: A group of one or more programming statements that has a name.
* int main (): Marks the beginning of a function.
* ;: This character marks the end of a complete programming statement.
* “ “: Encloses a string of characters, such as a message that is to be printed onto a screen.
* Variables: named storage locations that allow the programmer to store and access data in the computer’s RAM.
* Operators: Performs some operation on a piece of data (operand). e.g. ‘\*’ multiplies two operands; ‘=’ sets whatever is on the left of the operator equal to whatever is on the right.
* Lines: A single line that appears in the body of a program. Most of the time, lines contain something meaningful, sometimes they contain nothing at all
* Statements: A complete instruction that causes the computer to perform some action.
* cout: “console output”, object that displays plain text on a computer screen, simplest type of screen output.
* {}: Opening and closing braces. These braces contain the contents of a function.
* Variable definition: Defining all variables before they can be used, allowing the variables to be created in memory.
* Integers: Known as int, integers are whole numbers. Can be positive or negative. Commas cannot be  used even in very large numbers in C++.
* Floating point numbers: Any number with a decimal point. Can be positive or negative.
* C++ string class: Requires the preprocessor directive “#include <string>”. Must be enclosed in double quotation marks (i.e. “ ”).
* Identifiers: a programmer-defined name representing some element of a program. Cannot be any of the pre-defined C++ Key Words.
* Key Words: The “core” of the C++ language. All key words have specific pre-determined functions in C++.
* Strings/Literals/Constants: a value that cannot change during a program’s execution. Typically referred to as constants when they hold integers and literals when they hold strings.
* return 0;: Value that is returned If a program executes successfully.
* Punctuation: Plays an important role in programming. Syntax rules dictate where certain punctuation marks can and cannot appear. For example, a semicolon terminates a statement and is included at the end of most lines of code. However, a preprocessor directive cannot end with a semicolon because it is not a complete statement.
* Escape Sequences: ‘endl’ and …”/n”. When cout encounters the proper use of these sequences, the values are not placed onto the screen, however, the information that follows these sequences advances the output cursor to a new line. Additionally, endl will refresh the buffer.

***Testing Plan***

***Design***

**Design:**

Ask user for monthly interest rate, in decimal form

Input monthly interest rate

Ask user for total loan amount

Input total loan amount

Ask user for total number of payments

Input number of payments

Set monthly payment equal to [monthly interest rate times (1 plus monthly interest rate) to the power of monthly payments all divided by ((1 plus monthly interest rate) to the power of monthly payments) minus one] entirely multiplied by the total loan amount.

Set total paid back equal to monthly payment, times number of payments

Set interest paid equal to total paid back minus loan amount

Display monthly interest rate

Display total loan amount

Display number of payments

Display monthly payment

Display total amount paid back

Display interest paid