**CMSC 140 Programming Project 1**

**Chapters Covered:**

* Chapters 1, 2, and 3

**Concepts tested in this project**

* To work with cout and cin objects
* To work with variables, constants and literals
* To work with string object
* Use of different data types
* To follow programming style
* Use of basic arithmetic operators
* Use of output manipulators (setw, fixed, setprecision)

**Project Description**

The Department plans to purchase a humanoid robot. The Chairman would like you to write a program to show a greeting script the robot can use later. Your first task is to use the following script to prototype the robot for presentation:

|  |
| --- |
| *\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Robot Prototype Scripting \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  *Hello, welcome to Montgomery College! My name is Nao. May I have your name?*  ***john Smith***  *Nice to have you with us today, john Smith!*  *Let me impress you with a small game.*  *Give me the age of an important person or a pet to you.*  *Please give me only a number:*  ***2***  *You have entered 2.*  *If this is for a person, the age can be expressed as:*  *2 years*  *or 24 months*  *or about 720 days*  *or about 17280 hours*  *or about 1036800 minutes*  *or about 62208000 seconds.*  *If this is for a dog, it is 14 years old in human age.*  *If this is for a gold fish, it is 10 years old in human age.*  *Let's play another game, john Smith. Give me a whole number.*  ***4***  *Very well. Give me another whole number.*  ***5***  *Using the operator '+' in C++, the result of 4 + 5 is 9.*  *Using the operator '/', the result of 4 / 5 is 0*  *however, the result of 4.0 / 5.0 is about 0.8.*  *Thank you for testing my program!!*  *PROGRAMMER: Your Name*  *CMSC140 Common Project 1*  *Due Date:* |

Write a program that uses the script above as a guide without roles, i.e. robot computer and visitor human, to prototype robot greeting in C++. See the [Sample Screen Output](#SampleScreen) below.

**Project Specifications**

Input

* Visitor’s name
* An age
* Two numbers

Output: The program should display the following data:

* Complete script described above
* Your name as the programmer
* Assignment/Project number
* Due date

**Processing Requirements**

1. The program should declare and initialize (i.e., create and assign values for) variables and constants to hold (at least) the following data:

* Robot Name. This variable will hold the Robot Name. Initialize the variable with “Nao” or a name of your choice.
* Visitor Name, this variable will hold the user’s name.
* Age. This variable will hold a person’s or a pet’s age.
* A variable for Programmer’s Name. Initialize the variable with your full name.
* A constant variable for Assignment Number. Initialize the variable with the value 1.
* A constant variable for Due Date. Initialize the variable with the due date of this assignment.
* Declare Constant variables for
  + Days of Month. Initialize the variable with the value 30
  + Human Year. Initialize the variable with the value 1
  + Goldfish Year. Initialize the variable with the value 5
  + Dog’s age. Initialize the variable with the value 7
  + Examples how to declare constants:
    - const int ONE\_DOG\_YEAR = 7;
    - const int DAYS\_PER\_MONTH = 30;
  + Examples how to declare and initialize variables:
  + string programmer = "Kate Smiths"
  + string dueDate = "09/06/21";
  + int projectNum = 1;
  + double dogYear = 0;

1. Use the following data for the computations in the program :

* 1 month = 30 days
* Dog’s age = 7 times human’s age,
* Gold fish age = 5 times human’s age

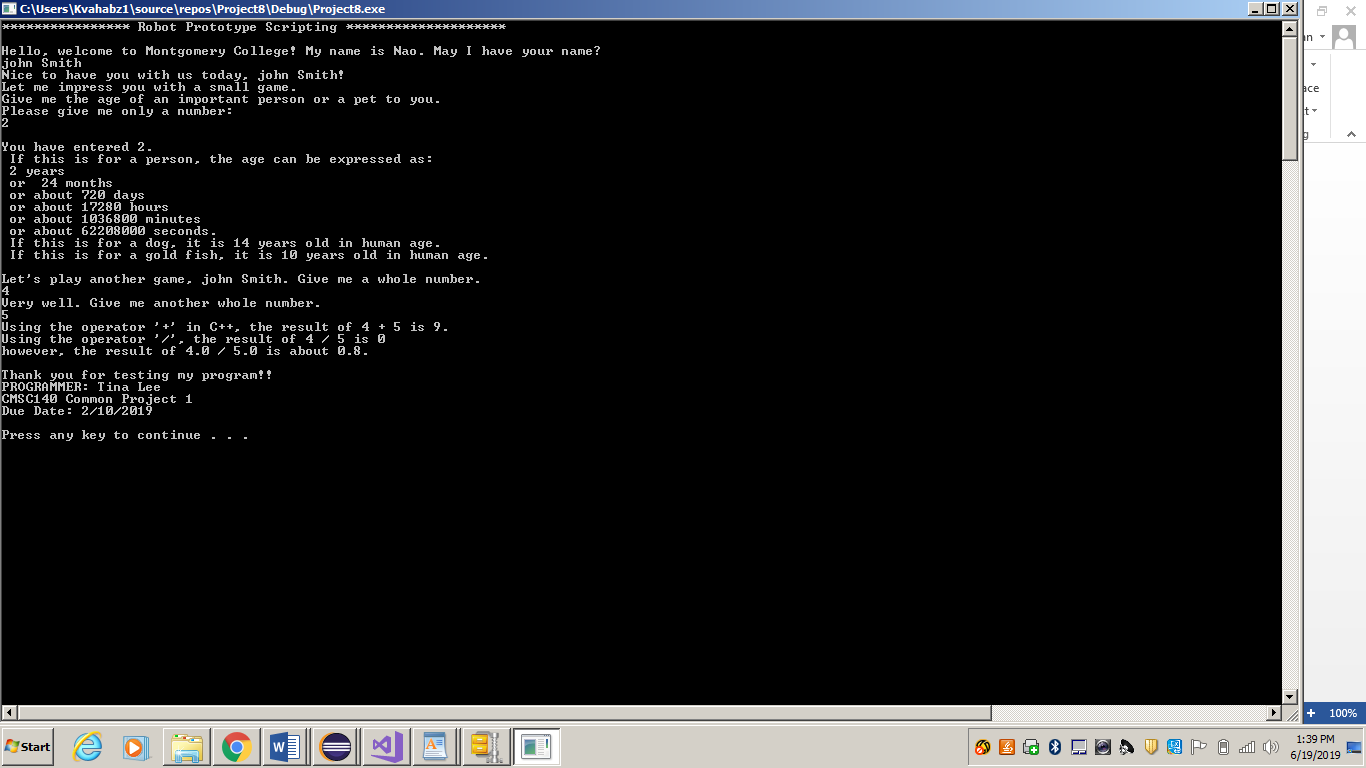
1. Use the above variables when creating the output of the program, for example:

cout << “My name is “ << robotName;

Where robotName is a variable defined in your program.

1. Your program should do all calculations using C++ expressions (Don’t use calculators for any calculations)
2. Use C++ constants to store all permanent data.

**Sample Screen Output**



**Test Plan**

Test your program with at least two more test cases. Use the given data as an example. Record your data for input and output in the following table. **Make sure your tests cover all the possible scenarios.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Case #** | **Input** | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | 2  4  5 | 24  720  17280  1036800  62208000  14  10  9  0  .8 |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |

**Check List**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr** |  | **Y/N** | **Comments** |
| **1** | **Source Code .cpp file:** FirstInitialLastNam\_Pr1.cpp | **Y** |  |
| **2** | **Compressed Source Code .cpp file:** FirstInitialLastNam\_Pr1.zip | **Y** |  |
| **3** | **Documentation File:** FirstInitialLastNam\_Pr1.cpp |  |  |
| **4** | **Program compiles** | **Y** |  |
| **3** | **Program runs with desired outputs related to a Test Plan** | **Y** |  |
| **5** |  | **Y** |  |
| **6** | **Documentation files:** | **Y** |  |
|  | Title page | **Y** |  |
|  | Comprehensive Test Plan | **Y** |  |
|  | Screenshots based on Test Plan | **Y** |  |
|  | Algorithm (Pseudocode) | **Y** |  |
|  | Flowchart | **Y** |  |
|  | Lessons Learned | **Y** |  |
|  | Student Check List | **Y** |  |

**CMSC140 Common Project Submission Requirements**

**Important NOTE:**

**-If your program does not compile, project will get a grade of “0”.** Contact your instructor prior to the project submission due date if you have compilation issues.

-If your project has compilation errors and you are not able to resolve them, try to see your instructor during his/her office hour or go to tutoring session for help **prior to project’s due date**.

**Details for Deliverables:**

1. Word document(docx) or .pdf with a name FirstInitialLastName\_Pr<number>.docx or .pdf should include:
   1. Title page including student name, project title and number
   2. Completed Test Plan table ( given in the project’s description document)
   3. Screenshot of the program run (using test data from the Test Plan)
   4. Pseudocode/Flowchart (if applicable to the project)
   5. Student Checklist
   6. Lessons Learned: Provide answers to the questions listed below:
      * + - Write about your Learning Experience, highlighting your lessons learned and learning experience from working on this project.
          - What have you learned?
          - What did you struggle with?
          - What would you do differently on your next project?
          - What parts of this assignment were you successful with, and what parts (if any) were you not successful with?
          - Provide any additional resources/links/videos you used to while working on this assignment/project.
2. FirstInitialLastName\_Pr1\_Moss.zip, a compressed file containing your C++ file (This folder should contain C++ source file FirstInitialLastName\_Pr1.cpp only).

Include the following in your code in C++ source file FirstInitialLastName\_Pr1.cpp:

Program Header: All programming projects’ source code need to have one block comment at the top of the program containing the course name and CRN, the project number, your name, project description, and the due date. Provide any additional comments as necessary to clarify the program.  
Following is a template of the required program header:

/\*

 \* Class: CMSC140 CRN

 \* Instructor:

\* Project<number>

 \* Description: (Give a brief description for Project)

 \* Due Date:

\* I pledge that I have completed the programming assignment independently.

   I have not copied the code from a student or any source.

   I have not given my code to any student.

   Print your Name here: \_\_\_\_\_\_\_\_\_\_

\* Pseudocode or Algorithm for the program:

(be sure to indent items with control structure)

(need to match flow chart submitted in documentation)

1.

2.

3.

(more as needed)

\*/

Comments: Add comment to variables, formulas, or any part of the program with the purpose of making the source code easier to understand.

Indentation: It must be consistent throughout the program and must reflect the control structure.

Proper naming conventions: Variable and method names need to be descriptive to show the role of the variable or method. Avoid single letter names. Constant names should be all upper case, variable names should use “camel case” (i.e. start with lower case, with subsequent words starting with upper case: *hours****W****orked* for example) or underscores to separate words (i.e. items\_ordered).

1. Checklist (grading sheet posted on Blackboard): Specific project checklist and grading rubric worksheet should be completed with your name in the highlighted cell and your initials in the checkboxes.

**CMSC140 Grading Rubric**

|  |  |
| --- | --- |
| **C140 Grading Rubric** | **Possible points deduct** |
| **TESTING** |  |
| Project must compile. If it doesn't compile | -100 |
| Project must run. If it's run- time error | -100 |
| Follows assignment document instructions | -25 |
| Doesn’t pass private instructor test | -75 |
| **Requirements** |  |
| Using integer or floating literals instead of constants or variables (for each literal -10 points) | -10 |
| Using calculator to calculate age in months, days, minutes instead of using C++ calculation expressions (– 20 points for each missing expression) | -20 |
| Missing static\_cast to cast int to double | -20 |
| **Documentation:** |  |
| Header comments at the top of the program are missing | -10 |
| Additional comments should be provided to clarify a code | -10 |
| Design: Pseudocode/algorithm missing or incorrect | -10 |
| Required output screenshots are missing | -10 |
| Lessons Learned are missing or incomplete | -10 |
| Student Checklist is missing or is not completed | -10 |
| FirstInitialLastName\_Pr1\_Moss.zip is missing | -100 |
| FirstInitialLastName\_Pr1.docx is missing | -50 |
| **Programming Style:** |  |
| Incorrect use of indentation, naming convention, etc: see coding/style standards | -10 |
| Output is not easy to understand | -5 |
| **Design:** |  |
| Does not print application header | -5 |
| Does not print the Programmer's name at the end | -5 |