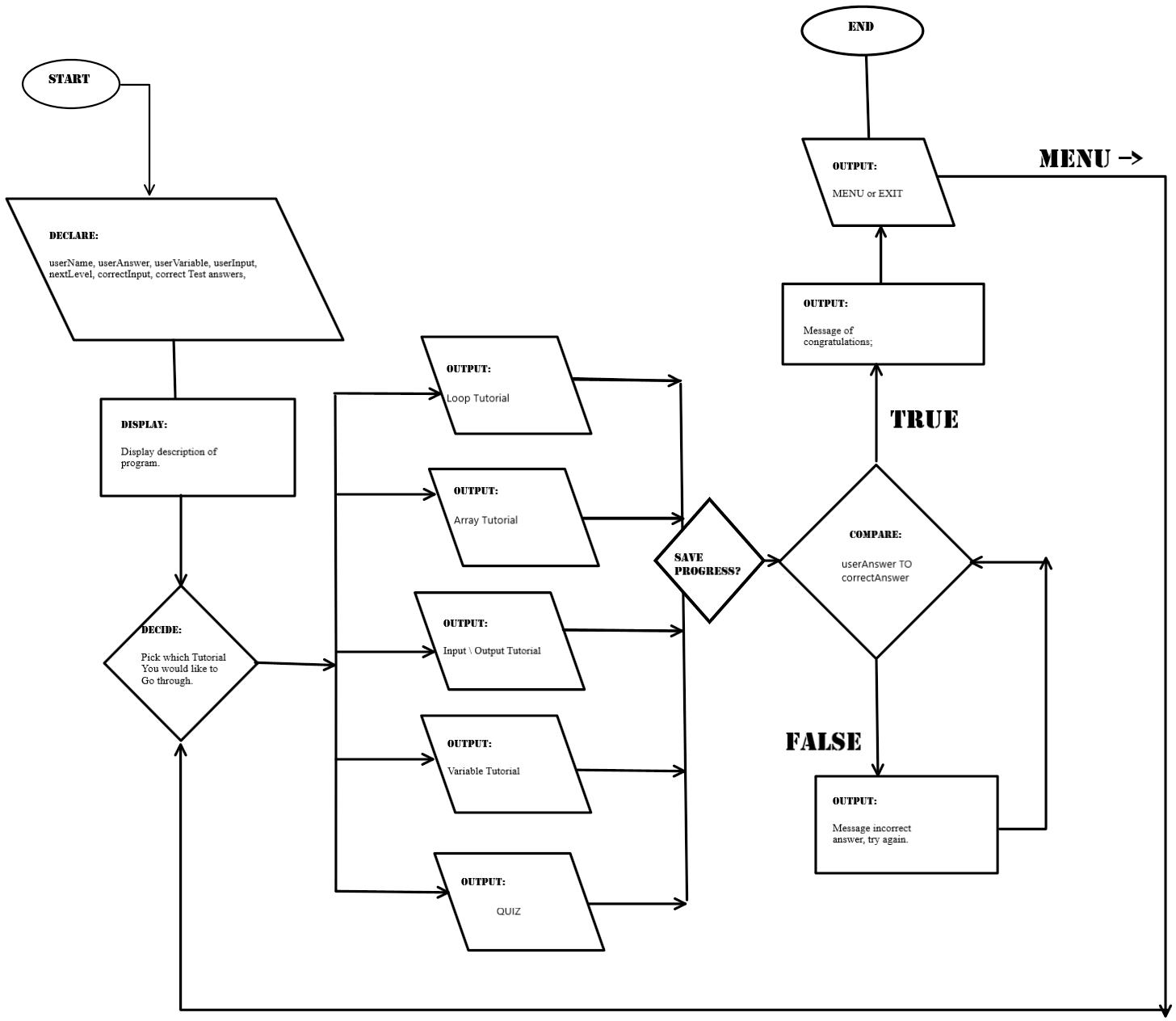
**PROJECT:** Programming Tutorial Program

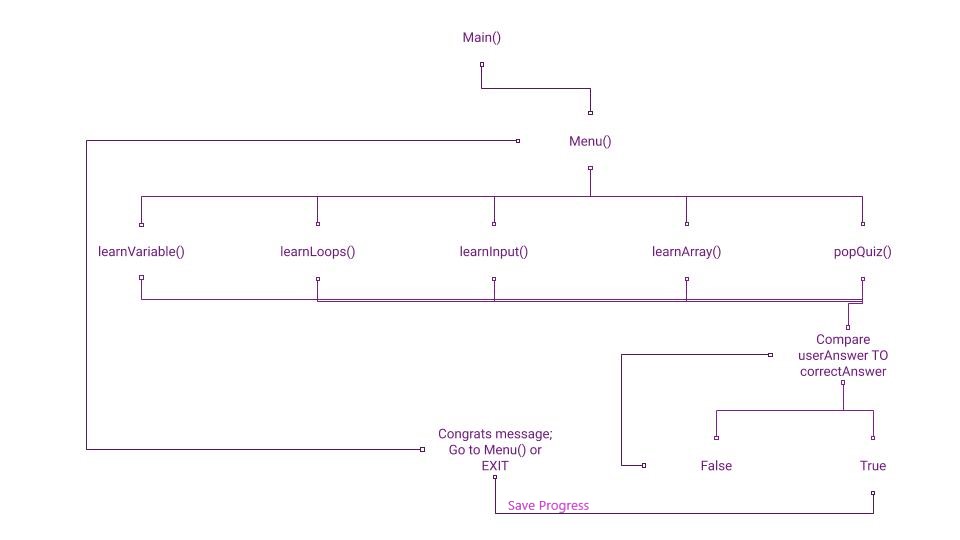
**IPO:**

|  |  |  |
| --- | --- | --- |
| **INPUT** | **PROCESS** | **OUTPUT** |
| userName, userAnswer, userVariable, userInput, correctInput, nextLevel, level1=”Input\Output”, level2=”Variable Declaration”, level3=”Loops”, level4=”Array”,testYourSkill, Correct Test Answers, | Comparing the user’s answer from the question to the correct answer.  Scoring the test  Save\Retrieve Progress | Correct answer = Move onto next question  Wrong answer = try question again  Quiz Grade  Save |

**Flowchart:**



**Hierarchy:**



**Pseudocode:**

**START**

**Declare:**

userName, userAnswer, letsGo, nextLevel, level1=”Input\Output”, level2=”Variable Declaration”, testYourSkill, correctAnswer, testScore, correctTestAnswers = { correctTest1, correctTest2, correctTest3, correctTest4, correctTest5 }, const userTestAnswers = { test1, test2, test3, test4, test5 }, Y, N, 1, 2;

**Display:**

“Good day! My name is Alan. What is yours?”

**Input:**

userName

**Display:**

“Welcome \_(userName)\_ to your Programming Tutorial! What would you like to learn today? A) Variable Declaration B) Input\Output C) Loops? Key in A, B, C?.”;

**Input:**

A || B || C

**Calculate:**

If userInput = A: run the Variable Declaration

If userInput = B: run the Input\Output lesson

If userInput = C: run the Loop lesson

**Display:***if 1*

“To start every program you have to declare your variables! You have to choose your data type, variable name and what you want your variable set to. In the end it will look something like this: string userName; Why don’t you give it a shot?”

**Input:**

userVariable

**Calculate:**

Compare the userAnswer to the correctAnswer →

If they match: **DISPLAY** “WooHoo, next level? Key in ‘Y’ or ‘N’”

If ‘Y’: run next question

If ‘N’: go back to menu

If they DO NOT match: **DISPLAY** “Uhoh! Try again” & loops back to try again.

**Display:**

“Would you like to save your progress? Key 1) Yes or 2) No.”

**Input:**

1 || 2

If 1: save

If 2: Move onto next mod

**Display:***if 2*

“Very basic in programming is the ability to accept input and print some output! Much like how you gave me your name. Here’s an example:

**Display:**

**cout << string userName;**

**cout << "What is your name?\n");**

**cout << std::cin >> userName;**

**cout << "Hello " << userName << " to your Programming Tutorial!" \n;**

**Display:**

“Ready to test your skill? Key ‘Y’ OR ‘N’ --> If N loop back to menu

**Display**:*if Y*

“You will be asked 5 questions about the following code:

“ int main()

{

1 int num1, num2, smallestNumber;

2 cout << "Welcome to the Smallest Number Guesser! \n";

3 cout << "You will be asked to enter two numbers.\n";

4 cout << "The smallest value will be displayed or a message if they are the same! \n";

5 cout << "Please, enter your first number: \n";

6 cin >> num1;

7 cout << "Please, enter your second number: \n";

8 cin >> num2;

9 if (num1 == num2)//if both numbers are equal

10 cout << "The numbers you entered are equal. Nice try!" << endl;

11 else if (num1 > num2)//if number 1 is bigger than number 2

12 smallestNumber = num2;//assign number 2 as smallest

13 else

14 smallestNumber = num1;//else assign number 1 as smallest

15 cout << "The smallest number is: " << smallestNumber << endl;

16 system("pause");

}

---------------------------------------------------------------------”

**Display:**

“Press ‘Y’ to begin.”

**Input:**

Y

**Display:**

“Which line contains the variables?”

**Input:**

test1

**Display:**

“Which line is the first place collecting input from the user?”

**Input:**

test2

**Display:**

“Which line is the last output of the program?”

**Input:**

test3

**Display:**

“How would you ask for the user’s name?”

**Input:**

test4

**Display:**

“What type of loop executes code before checking the condition?”

**Input:**

test5

**Calculate:**

Compare userAnswer1, userAnswer2, userAnswer3, userAnswer4, userAnswer5 to the correct answers and grade the test.

**Display:**

“Here is your score: “ << testScore << endl;

**Display:**

“Would you like to save your progress? Key 1) Yes or 2) No.”

**Input:**

1 || 2

If 1: save

If 2: Move onto next mod

**Display:**

“Would you like to 1.) try again or 2.) end program?”

**Input:**

1 || 2

If 1: start test again

If 2: end program & **DISPLAY:** Goodbye!

**Display:**

Loop Lesson→

“Differences between for, do\while, while loops”.

**Display:**

“for (i = 1; i <= 10; i++) {

printf( "Hello World\n");

}

while (i < 6) {

printf( "Hello World\n");

// update expression

i++;

}

do

{

// loop body

printf( "Hello World\n");

// update expression

i++;

} while (i < 1); // test expression”

**Display:**

“If you wanted to test your conditions before your code runs, what kind of loop are you going to run?”

**Input:**

correctLoop

**Display:**

“Would you like to save your progress? Key 1) Yes or 2) No.”

**Input:**

1 || 2

If 1: save

If 2: Move onto next mod

**Display:**

“Would you like to 1.) try again or 2.) end program?”

**Input:**

1 || 2

If 1: try again

If 2: end program & **DISPLAY:** Goodbye!

**Display:**

Array Lesson→

“Let;s Get familiar with Arrays”.

**Display:**

“Say you wanted to put a list of your top 3 fav cat names together for your new pet.”

“Void main() {

string catNames[3] = {“”, “”, “”, “”, “”, “”};

]”

“Inside the double quotes you would put the names you have chosen”

“Give it a shot and write out an empty array.”

**Input:**

correctArray

**Display:**

“Would you like to save your progress? Key 1) Yes or 2) No.”

**Input:**

1 || 2

If 1: save

If 2: Move onto next mod

**Display:**

“Would you like to 1.) try again or 2.) end program?”

**Input:**

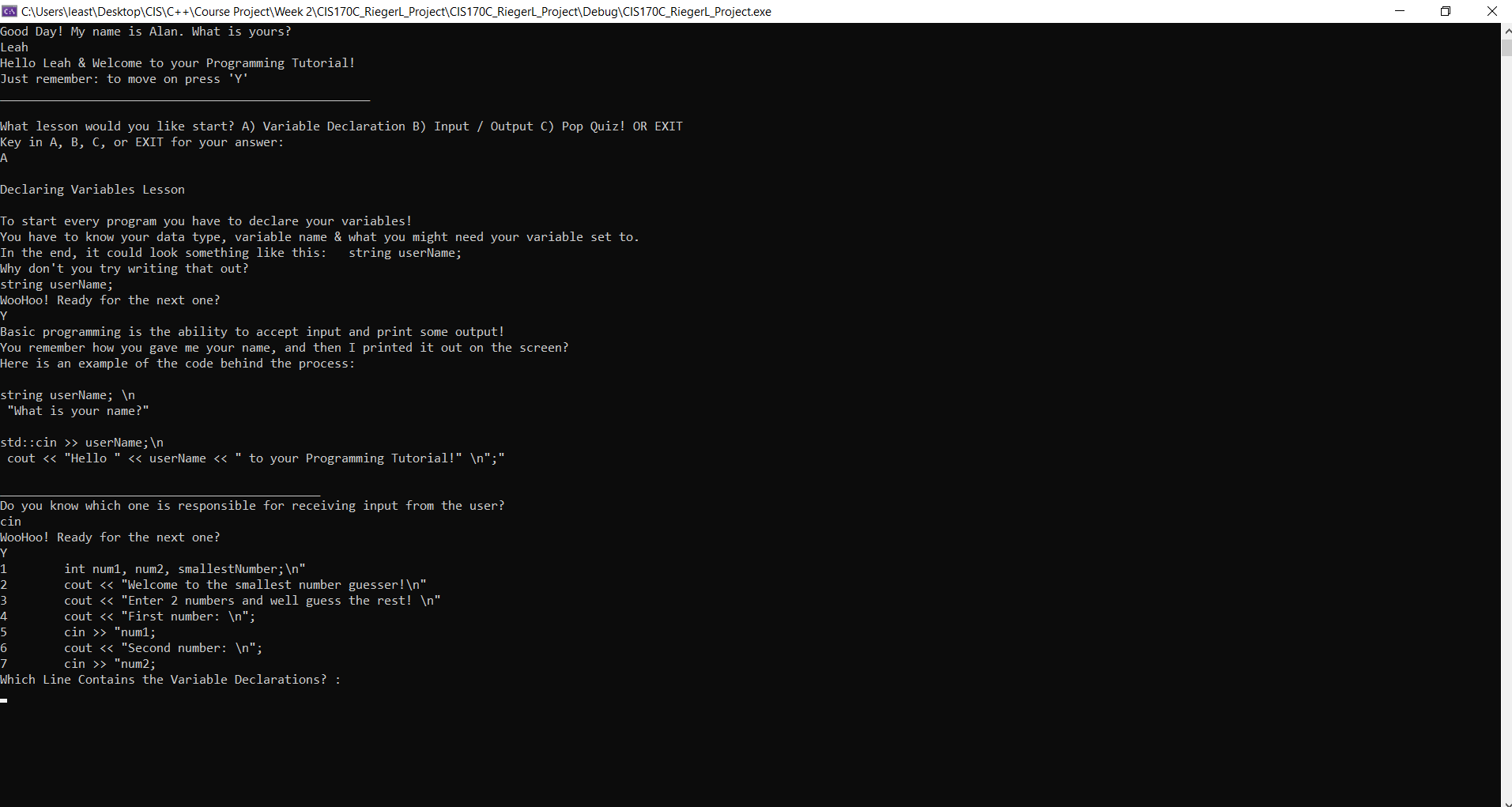
1 || 2

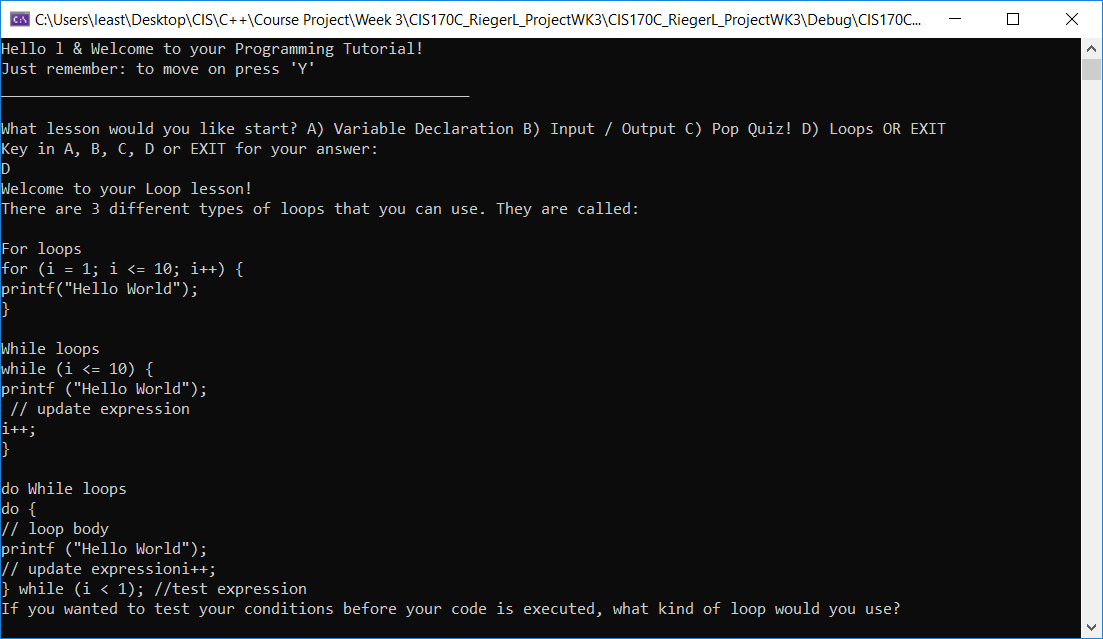
If 1: try again

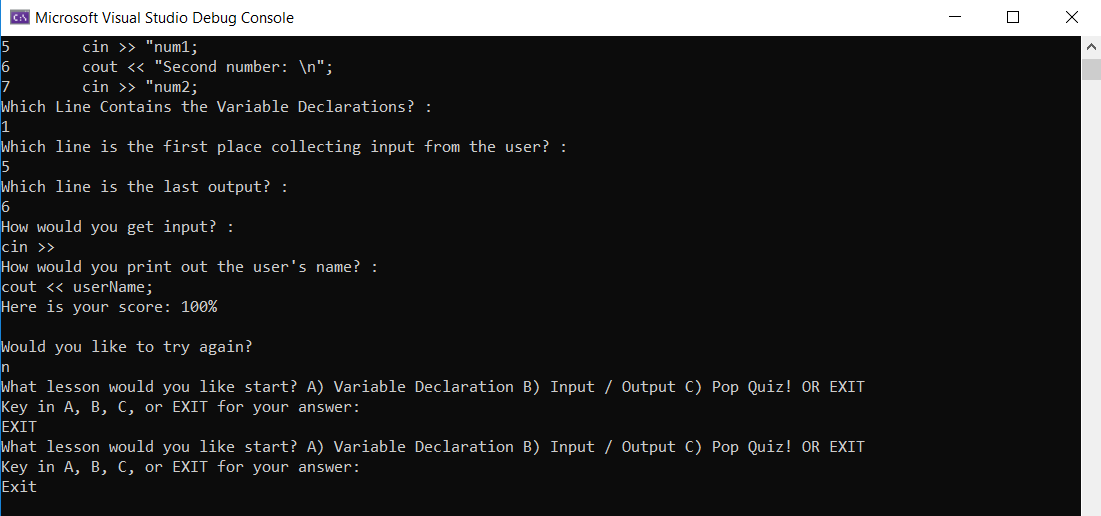
If 2: end program & **DISPLAY:** Goodbye!

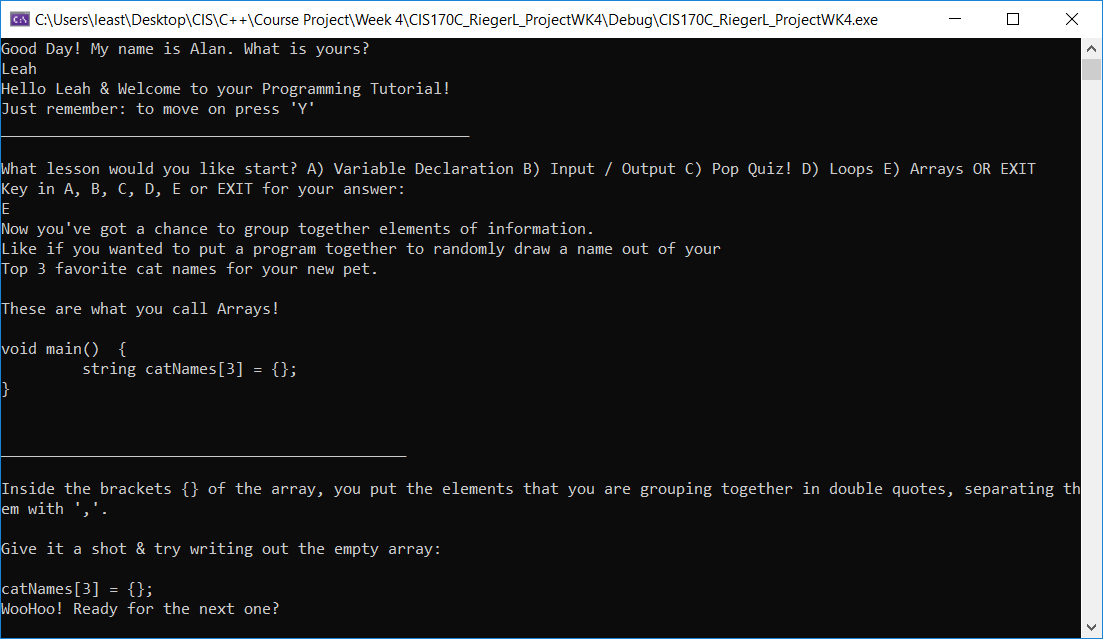
**STOP**

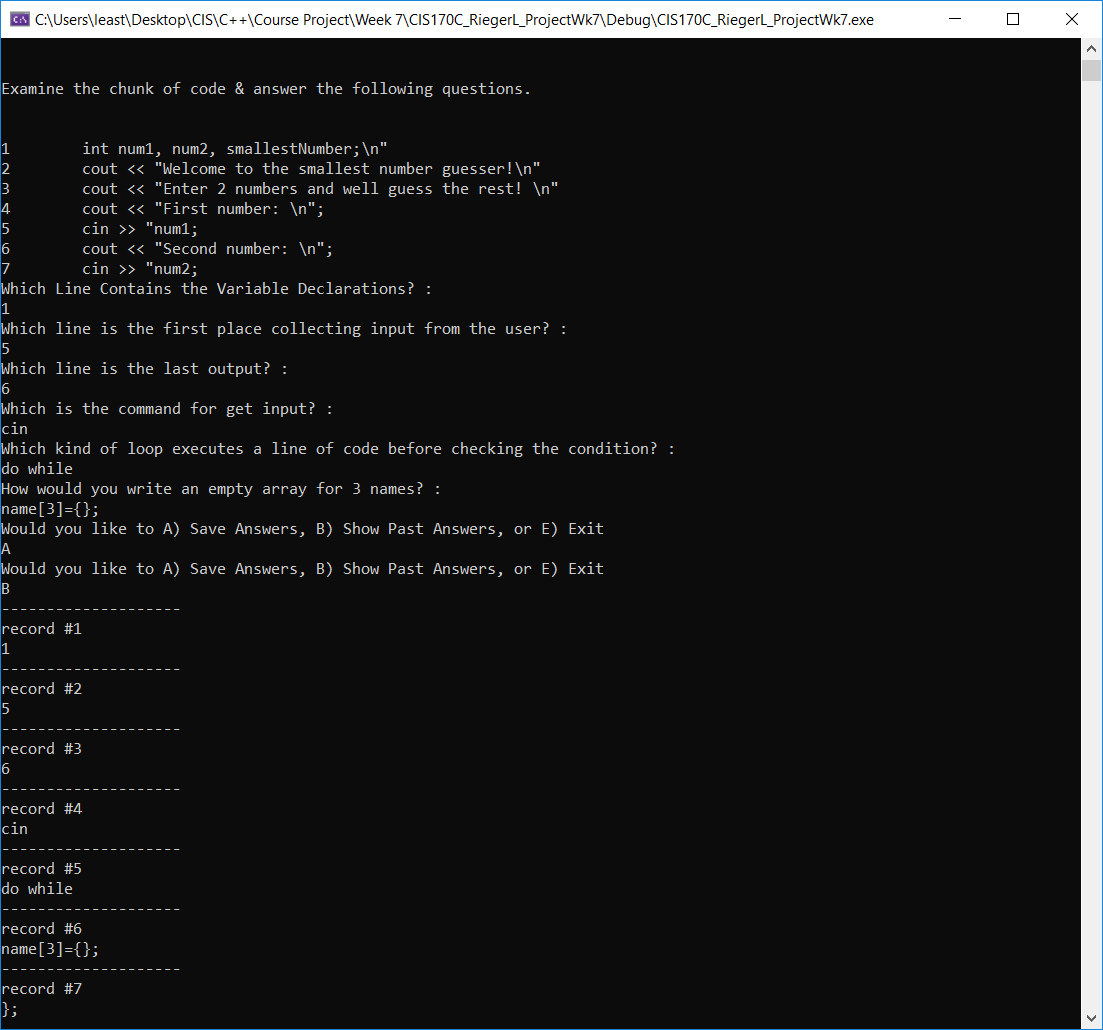
**SNAPSHOT:**

****

****

****

****

****

**CODE:**

**Source:**

// ---------------------------------------------------------------

// Programming Assignment: Course Project

// Developer: Leah Rieger

// Date Written: 08/23/2019

// Purpose: Programming Tutorial Program

// ---------------------------------------------------------------

#include <iostream>

#include <string>

#include <cstdlib>

#include "Project.h"

using namespace std;

int main() {

MyClass program;

program.mainMenu();

return 0;

system("pause");

}

**Header:**

#pragma once

// ---------------------------------------------------------------

// Programming Assignment: Course Project

// Developer: Leah Rieger

// Date Written: 08/23/2019

// Purpose: Programming Tutorial Program

// ---------------------------------------------------------------

#pragma once

#include <iostream>

using namespace std;

class MyClass {

public:

MyClass();

void mainMenu();

void selection(string userAnswer);

void learnVariables();

void learnInput();

void learnLoops();

void learnArray();

void popQuiz();

void levelUp(string correctAnswer, string userVariable);

void testScores(int numberCorrect);

void endOfProgram(string finalAnswer);

string getUserName();

void setUserName(string userName);

void writeData(string arrayOfStringsToSave [6], string FileName);

void readData(string FileName);

private:

string userName;

};

**.CPP:**

// ---------------------------------------------------------------

// Programming Assignment: Course Project

// Developer: Leah Rieger

// Date Written: 08/23/2019

// Purpose: Programming Tutorial Program

// ---------------------------------------------------------------

#include <iostream>

#include <fstream>

#include "Project.h"

#include <string>

#include <cstdlib>

MyClass::MyClass() {

//Begin the description of the program & what lesson to go to first

string userName;

cout << "Good Day! My name is Alan. What is yours? \n";

std::cin >> userName;

setUserName(userName);

cout << "Hello " << userName << " & Welcome to your Programming Tutorial!\n";

cout << "Just remember: to move on press 'Y'\n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " \n";

}

string MyClass::getUserName() {

return userName;

}

void MyClass::setUserName(string userName1) {

userName = userName1;

}

//Main Menu function - will ask user what part of program they want to complete

void MyClass::mainMenu() {

string userAnswer;

cout << "What lesson would you like start? A) Variable Declaration B) Input / Output C) Pop Quiz! D) Loops E) Arrays OR EXIT\n";

cout << "Key in A, B, C, D, E or EXIT for your answer: " << endl;

std::cin >> userAnswer;

selection(userAnswer);

}

//Selection function - will go to what function the user has selected

void MyClass::selection(string userAnswer) {

//if usr a\b\c execute function

if (userAnswer == "A") {

learnVariables();

}

if (userAnswer == "B") {

learnInput();

}

if (userAnswer == "C") {

popQuiz();

}

if (userAnswer == "D") {

learnLoops();

}

if (userAnswer == "E") {

learnArray();

}

if (userAnswer == "EXIT") {

mainMenu();

}

}

//Function to compare users input to correct answer & move onto next chapter of tutorial

void MyClass::levelUp(string correctAnswer, string userVariable) {

//loop to continue when entering a wrong answer

string nextLevel;

while ((correctAnswer != userVariable))

{

cout << "Uhoh!! That's not quite it, let's try again. \n";

std::getline(cin, userVariable);

};

//Processing, Comparing answers

if (correctAnswer == userVariable) {

cout << "WooHoo! Ready for the next one?\n"; // if no & return need to call menu function

std::cin >> nextLevel;

}

}

//First function - goes to learning how to declare a variable

//Based on users answer to run the variable declaration program

void MyClass::learnVariables() {

const string correctAnswer = "string userName;";

string userVariable;

cout << " \n";

cout << "Declaring Variables Lesson\n";

cout << " \n";

cout << "To start every program you have to declare your variables!\n";

cout << "You have to know your data type, variable name & what you might need your variable set to.\n";

cout << "In the end, it could look something like this: \t string userName; \n";

cout << "Why don't you try writing that out?\n";

// A new line character(/n) keeps getting inserted into cin and continuing past the getline (because of a bug where std::cin >> leaves a /n in the cin buffer).

// We use Ignore to clear any previous input.

cin.ignore();

// Get line allows free entry of text up to a delimiter. If one is not specified it defaults to /n.

getline(cin, userVariable);

levelUp(correctAnswer, userVariable);

learnInput();

}

//Learning how to take input from a user and how to print your output

void MyClass::learnInput() {

const string correctInput = "cin", correctOutput = "cout";

string userInput;

cout << "Basic programming is the ability to accept input and print some output!\n";

cout << "You remember how you gave me your name, and then I printed it out on the screen?\n";

cout << "Here is an example of the code behind the process: \n";

cout << " \n";

cout << "string userName; \\n\n";

printf(" \"What is your name?\"\n\n");

cout << "std::cin >> userName;\\n\n";

printf(" cout << \"Hello \" << userName << \" to your Programming Tutorial!\" \\n\";\"\n");

cout << " \n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << "Do you know which one is responsible for receiving input from the user? \n";

cin.ignore();

getline(cin, userInput);

levelUp(correctInput, userInput);

learnLoops();

}

//user learns the different types of loops

void MyClass::learnLoops() {

string loopAnswer;

const string correctLoop = "do While loops";

cout << "Welcome to your Loop lesson!\n";

cout << "There are 3 different types of loops that you can use. They are called: \n";

cout << " \n";

cout << "For loops \n";

cout << "for (i = 1; i <= 10; i++) { \n";

cout << "printf(\"Hello World\"\); \n";

cout << "} \n";

cout << " \n";

cout << "While loops \n";

cout << "while (i <= 10) { \n";

cout << "printf (\"Hello World\"\); \n";

cout << " // update expression \n";

cout << "i++; \n";

cout << "} \n";

cout << " \n";

cout << "do While loops \n";

cout << "do { \n";

cout << "// loop body \n";

cout << "printf (\"Hello World\"\); \n";

cout << "// update expression";

cout << "i++; \n";

cout << "} while (i < 1); //test expression \n";

cout << "If you wanted to test your conditions before your code is executed, what kind of loop would you use? \n";

cin.ignore();

getline(cin, loopAnswer);

levelUp(correctLoop, loopAnswer);

learnArray();

}

//Getting Familiar with Arrays.

void MyClass::learnArray() {

const string correctInput = "catNames[3] = {};";

string userInput;

cout << "Now you've got a chance to group together elements of information.\n";

cout << "Like if you wanted to put a program together to randomly draw a name out of your\n";

cout << "Top 3 favorite cat names for your new pet.\n";

cout << " \n";

cout << "These are what you call Arrays!";

cout << " \n";

cout << " \n";

cout << "void main() { \n";

cout << "\t string catNames[3] = {};\n";

cout << "}\n";

cout << " \n";

cout << " \n";

cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

cout << " \n";

cout << "Inside the brackets {} of the array, you put the elements that you are grouping together in double quotes, separating them with ','.\n";

cout << " \n";

cout << "Give it a shot & try writing out the empty array: \n";

cout << " \n";

cin.ignore();

getline(cin, userInput);

levelUp(correctInput, userInput);

popQuiz();

}

//Gives the user a pop quiz on the information they have gone over

void MyClass::popQuiz() {

string FileName = "TestAnswers.txt";

char userChoice = ' ';

//variables

int i, numberCorrect = 0;

const int SIZE = 6;

string userTestAnswers[SIZE];

const string testQuestions[] = { "Which Line Contains the Variable Declarations?", "Which line is the first place collecting input from the user?", "Which line is the last output?", "Which is the command for get input?", "Which kind of loop executes a line of code before checking the condition?", "How would you write an empty array for 3 names?" };

const string correctTestAnswers[] = { "1", "5", "6", "cin", "do While", "names[3] = {};" };

cout << "\tPOP QUIZ TIME\n";

cout << " \n";

cout << " \n";

cout << "Examine the chunk of code & answer the following questions.\n";

cout << " \n";

cout << " \n";

cout << "1\t int num1, num2, smallestNumber;\\n\"\n";

cout << "2\t cout << \"Welcome to the smallest number guesser!\\n\"\n";

cout << "3\t cout << \"Enter 2 numbers and well guess the rest! \\n\"\n";

cout << "4\t cout << \"First number: \\n\";\n";

cout << "5\t cin >> \"num1;\n";

cout << "6\t cout << \"Second number: \\n\";\n";

cout << "7\t cin >> \"num2;\n";

cin.ignore();

//Get test answers

for (i = 0; i < 6; i++) {

cout << testQuestions[i] << " : " << endl;

getline(cin, userTestAnswers[i]);

if (userTestAnswers[i] == correctTestAnswers[i]) {

numberCorrect += 1;

}

}

//Add ability to save to disk

do {

//ask for user's choice

std::cout << "Would you like to A) Save Answers, B) Show Past Answers, or E) Exit\n";

std::cin >> userChoice;

userChoice = toupper(userChoice);

if (userChoice == 'A') {

writeData(userTestAnswers, FileName);

}

if (userChoice == 'B') {

readData(FileName);

}

} while (userChoice != 'E');

testScores(numberCorrect);

}

//This one takes user's answers, compares them to the correct answers & grades

void MyClass::testScores(int numberCorrect) {

string finalAnswer;

int finalScore = 0;

finalScore = (100 / 6) \* numberCorrect;

cout << "Here is your score: " << finalScore << "% \n" << endl;

cout << "Would you like to try again?\n";

cin.ignore();

getline(cin, finalAnswer);

endOfProgram(finalAnswer);

}

//This writes to a file

void MyClass::writeData(string arrayOfStringsToSave [6], string FileName) {

ofstream outMyStream(FileName, ios::in);

for (int i = 0; i < 6; i++) {

//'#' is the delimiter

outMyStream << arrayOfStringsToSave[i] << "#";

}

outMyStream.close();

}

//This reads from a file

void MyClass::readData(string FileName) {

ifstream inMyStream(FileName);

if (inMyStream.is\_open()) {

//set delimiter

string recBreaks = "";

recBreaks.assign(20, '-');

int fieldCount = 0; // num fields read

int recordCount = 1; // num records read

//read first field

fieldCount = 1;

string fieldBuffer;

while (!inMyStream.eof()) {

// display the field

switch (fieldCount) {

case 1:

std::cout << recBreaks << endl;

std::cout << "record #" << recordCount << endl;

fieldCount = 0;

recordCount++; break;

}

//read next

std::getline(inMyStream, fieldBuffer, '#');

std::cout << fieldBuffer << endl;

fieldCount++;

}

std::cout << recBreaks << endl;

inMyStream.close();

}

}

//This wraps up the program & gives option to go back to main menu

void MyClass::endOfProgram(string finalAnswer) {

if (finalAnswer == "Y") {

popQuiz();

}

if (finalAnswer != "N") {

mainMenu();

}

}