Programming Fundamentals I

COSC1436--Fall 2015

Lab 4. Conditional Statements

Objectives:

1. To work with relational and logical operators
2. To work with conditional statements (if. if-else, if-else-if, nested if-else, and switch statements)
3. To practice creating flow charts of conditional statements
4. To practice analyzing code by hand tracing code and/or its flow chart
5. To learn to generate a test data set from problem specifications
6. To practice using flow charts in designing and verifying algorithms with conditional statements.

Make sure to demo your work to your instructor for each task to get credit. You also need to submit the test data set and flowchart for each task.

Task 1: Working with Conditional Statements--Analysis

Exercise 1. Create a flow chart for the following program (use the back of this page or staple a separate sheet):

```
// This program displays course grade based on the average score
// using the standard grading formula:
// average ≥ 90 → A;  80 ≤ average < 90 → B;  70 ≤ average < 80 → C ;
// 60 ≤ average < 70 → D; average < 60  → F
// PLACE YOUR NAME HERE
#include <iostream>
using namespace std;

int main()
{
	float avgScore; // holds the grade average
	char  grade;              // holds a letter grade

	cout << "Input your average:" << endl;
	cin >> avgScore;

	if (avgScore >=90)
		grade ='A';
	if (avgScore >= 80)
		grade = 'B';
	if (avgScore >= 70)
		grade = 'C';
	if (avgScore >= 60)
		grade = 'D';
	else
		grade = 'F';

	cout << "Your grade is: " << grade << endl;
	system("pause");
	return 0;
}
```

Exercise 2. Manually trace your flow chart and/or the code to predict the program output for each of the following inputs:

Input	Expected output
55	F
65	D
75	C
95	A

Exercise 3. Based on your analysis of the above program, list the range of average score values for each letter grade. Does the program meet its specifications as stated in the documentation (see first comments lines above program code)?

Grade	Average score range
A	90 – 100
B	80 – 90
C	70 – 80
D	60 – 70
F	0 – 60

Exercise 4. Create a new C++ project in Visual Studio, add a new C++ file and copy the above program code into it.

Exercise 5. Test your program and verify your answers to exercises 3 and 4.

Demo your answers to exercise 3 & 4 to your instructor. Submit your flowchart to D2L Dropbox.

Task 2: Working with Conditional Statements—Design

In this task you're to fix the program of the previous task.

Exercise 1. Based on the program description in its documentation (see few comment lines at begin of program), create a test data set for verification and testing purposes.

Exercise 2. Create a flow chart based on the program description. Verify your flowchart with the test data set.

Exercise 3. Create a new project (or modify the old one from Task1) and write the code for the flowchart. Test your program using your test data set.

Demo your answers to exercise 3. Submit your test data set and flowchart to D2L dropbox

Task 3: If-else-if statements and Logical Operators--Analysis

Exercise 1. Consider the following program:

```
// This program illustrates the use of logical operators
// PLACE YOUR NAME HERE

#include <iostream>

using namespace std;

int main()
{
      char employmentStatus;
      float performanceScore, salaryRaise;

      cout << "What's your employment status?" << endl;
      cout << "Enter 1 (full-time), 2 (part-time), or 3 (temporary)"
            << endl << endl;

      cin >> employmentStatus;

      cout << "Now enter your performance score" << endl;

      cin >> performanceScore;

      if (performanceScore >= 8.0 && employmentStatus == '1')
            salaryRaise = 1000.0;

      else if (employmentStatus != '1'|| performanceScore >= 6.0)
            salaryRaise = 150.0;

      else if (employmentStatus == '3')
            salaryRaise = 0;

      cout << "Your salary raise is: $" << salaryRaise<<endl;


      system("pause");
      return 0;
   }
```

Analyze the above program and predict its output in each of the following cases (do NOT run the program, predict its behavior using code walk through):

- full-time employee with performance above 8.0? 1000
- part-time employee with performance above 7.0? 150
- part-time employee with performance above 5.0? 0
- temporary employee with performance above 7.0? 0
- Does the code take care of all possible cases i.e. are there cases for which the program will not generate a salary raise? No

Exercise 2. List all possible cases and select a sample input and expected output for each case.

Exercise 3. Copy the program in a new project. Test the program using the test data set you generated in the previous exercise._Report and explain any discrepancies between expected and obtained results.

Demo your program. Make sure to have your test data set ready for the demo.

Task 4: If-else-if statements and Logical Operators--Design

Modify the program of the previous task to achieve the following results. Use logical operators and if-else-if statements.

- full-time employees get $500.00 raise plus 4% of their current salary if their performance score is 8 or above and 2.5% if their performance score is below 8 but greater or equal to 6
- part-time employees with performance score of 8 or above get a raise equal to 3% of their current salary while those with a performance score below 8 but greater or equal to 6 get a 1.5% raise.
- no raises are given in any other cases.

Exercise 1. Based on the above specifications, list all the possible cases of employee status and performance score. Generate a test data set.

Exercise 2. Create a flowchart that meets the above specifications. Verify your algorithm using the test data set generated in the previous exercise.

Exercise 3. Modify your project from the previous task based on your algorithm. Test your project using your test data set. Does your program meet the problem specifications? _____

Submit your flowchart and test data set to D2L dropbox. Demo your program. Make sure to have your test data set ready for the demo.

Task 5: Algorithm design using conditional statements

Exercise 1. Enter the code below in a new C++ project and run it (or copy and paste from file menuDrivenInterface.txt on D2L). The program is an example of how switch statements can be used to build a menu driven interface. Remove the break statement at the end of each case. What is the effect on the execution of the program?

```
// This program illustrates the use of the switch statement for a menu driven interface.
// PLACE YOUR NAME HERE
#include <iostream>
using namespace std;

int main()
{
	const double ADULT = 40.0, CHILD = 30.0, SENIOR = 15.0; // Club membership monthly costs
	int months;		// length of membership
	int choice;

	cout << "Please select one of the membership options below: " << endl;
	cout << '\t'<< "1.  Adult Membership"<<endl;
	cout << '\t'<< "2.  Child Membership"<<endl;
	cout << '\t'<< "3.  Senior Citizen Membership"<<endl;
	cout << '\t'<< "4.  Quit the program"<<endl<<endl;
	cout << "Enter your choice: ";
	cin >> choice;

	switch( choice ) // This is where the switch statement begins
	{
		case 1: cout << "\n For how many months? "<< endl;
				cin >> months;
				cout <<"Your total membership cost is: $"<< months * ADULT << endl;
				 break;
		case 2: cout << "\n For how many months? "<< endl;
				cin >> months;
				cout <<"Your total membership cost is: $"<< months * CHILD << endl;
				 break;
		case 3: cout << "\n For how many months? "<< endl;
				cin >> months;
				cout <<"Your total membership cost is: $"<< months * SENIOR <<
endl;
				 break;
		case 4: cout << "No membership was selected....";
					cout <<"Good bye!"<<endl;
				break;
		default: cout << "The valid choices are 1 though 4" << endl;
				 cout <<"Run the program again and make a valid selection"<<endl;
	}

	system("pause");
	return 0;
}
```

Exercise 2. Add an option to your program for a family membership (monthly cost $60)

Demo your program.

Task 6: Algorithm design using conditional statements

Exercise 3. Complete the following program so that it reads three test scores from the standard input and displays the largest of the three on the screen monitor.

```
// This program finds the largest of three test scores and
// displays it.
// PLACE YOUR NAME HERE

#include <iostream>

using namespace std;

int main() {

	float testScore1, testScore2, testScore3;

	cout <<"Enter score for test 1: " << endl;
	cin >> testScore1;
	cout << "Enter score for test 2: " << endl;
	cin >> testScore2;
	cout <<"Enter score for test 3: " << endl;
	cin >> testScore3;

		/*  To BE FILLED  */

	return 0;
}
```

Exercise 2. List the possible case scenarios that should be considered and create a test data set. Example of cases to be considered: test score 1 is the largest; test score 2 is largest, etc. You should also consider the cases when scores are equal.

Exercise 3. Write an algorithm using conditional statements to find the largest of three values. Draw a flowchart of your algorithm and verify it with your test data from previous exercise.

Exercise 4. Create a project in Visual Studio and fill in the missing code in the program above based on your algorithm of the previous exercise. Test your program using your test data.

Submit your flowchart and test data set to D2L dropbox. Demo your program. Make sure to have your test data set ready for the demo.