

The American University in Cairo
Computer Science & Engineering Department
CSCE 106/1001 – Fundamentals of Computer Science

Lab Session 7

Programming of Multiple Selections

Objectives:

- To learn how to develop algorithm for multiple selections.
- To learn how to implement multiple decisions in algorithms using the **nested-if** statement.
- To learn how to implement multiple decisions in algorithms using the **switch** structure.

Problems:

1. At AUC, the academic standing of a student is determined based on the number of the credit hours that the student has achieved. The rules are as follows:

Number of Credit Hours (CrH)	Academic Standing
CrH < 30	Freshman
30 <= CrH < 60	Sophomore
60 <= CrH < 90	Junior
90 <= CrH < 120	Senior
120 <= CrH	Graduating Senior

Draw a flow chart and write a program in C++ to input the number of credit hours achieved by the student and print out his/her academic standing. **Implement your solution in C++ once using nested-if structure and second using switch structure. Validate the user input to accept only number of credit hours that is between 0 and 133 (inclusive).**

2. The island nation of Babbage charges its citizens an income tax each year. The tax rate is based on the following table. Write a C++ program that, when given a person's income, prints the tax owed rounded to the nearest dollar.

Income	Tax Rate
\$ 0-5,000	0
\$ 5,001-10,000	3%
\$ 10,001-20,000	5.5%
\$ 20,001-40,000	10.8%
Over \$ 40,000	23.7%

3. Your Professor of CSCE 1001, wants you to write a program to compute and print the grade for a student in his class. The grade is based on three mid-term examinations (worth a possible 100 points each) of which the best two are only considered in the computation, five quizzes (10 points each), and a 200-point final examination. Your output should include all scores, the percentage grade, and the letter grade. The grading scale is:

Percentage Grade (PG)	Letter Grade
90 <= PG <= 100	A
80 <= PG < 90	B
70 <= PG < 80	C
60 <= PG < 70	D
0 <= PG < 60	F

Instructions:

- The TA solves problem 1 (**30 minutes**)
- The TA asks students to solve and **submit in-Lab** problem 2 (**30 minutes**)
- The TA asks students to solve problem 3 and **submit before the next Lab session**