ENGR1200U Introduction to Programming for Engineers (Winter 2013)

Assignment 2: Selection & Repetition Control Structures Using C++ (20 marks)

Due Date: By 5:30pm on Tuesday, Mar 5, 2013

The purpose of this assignment is to practice your understanding of selection and repetition control structures by designing and developing C++ programs that are more involved than those in Assignment#1. You are required to answer all questions. Write a separate C++ program to solve each of the three problems. Submit a hard copy of all your source code as per the submission instructions section below. **This is an individual work assignment.**

IMPORTANT NOTE: No advanced features (e.g. user functions, arrays, etc.) that have not been covered in class yet are allowed to be used in your programs. Students with solutions that use advanced features will lose a significant number of marks (if not all) for that solution.

1. *[Speeding Car - 6 marks]* A red and blue car were involved in a head-on collision. The red car was at a standstill and the blue car was possibly speeding. Eyewitness video recorded immediately following the moment of impact showed the two cars entangled together, traveling at 30 mph. The red car weighs 2,000 pounds, and the speed limit is 65 mph. Write a C++ program that allows the user to enter the weight of the blue car and determines whether the blue car was speeding.

   Hint: 1 mph = 0.447 m/s, and 1 pound = 0.4536 kg. Conservation of momentum dictates that the total momentum before a collision equals the total momentum after a collision as in the following equation:

$$m_1 v_1 + m_2 v_2 = (m_1 + m_2) v$$

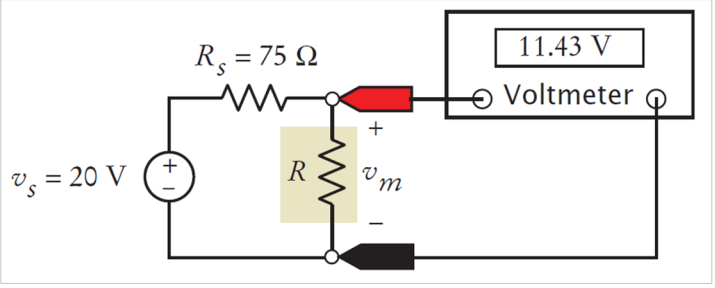
2. *[Credit Card Number – 6.5 marks]* The last digit of a credit card number is the check digit, which protects against transcription errors such as an error in a single digit or switching two digits. The following method is used to verify actual card numbers but, for simplicity, we will describe it for numbers with 8 digits instead of 16:
   a. Starting from the rightmost digit, form the sum of every other digit. For example, if the credit card number is 43589795, then you form the sum 5 + 7 + 8 + 3 = 23.

b. Double each of the digits that were not included in the preceding step. Add all digits of the resulting numbers. For example, with the number given above, doubling the digits, starting with the next-to-last one, yields 18 18 10 8. Adding all digits in these values yields 1 + 8 + 1 + 8 + 1 + 0 + 8 = 27.

c. Add the sums of the two preceding steps. If the last digit of the result is 0, the number is valid. In our case, 23 + 27 = 50, so the number is valid.

Write a C++ program that implements this algorithm. The user should supply an 8-digit number, and you should print out whether the number is valid or not. If it is not valid, you should print out the value of the check digit that would make the number valid. The program should give the user the option whether to continue with checking more credit card numbers (i.e. entering letter 'C') or exit the program (i.e. entering the letter 'E').

3. *[Meter Voltage - 7.5 marks]* The figure below shows an electric circuit designed to measure the temperature of the water in a beaker.



The resistor *R* represents a temperature sensor enclosed in the beaker. The resistance *R*, in Ω, is related to the temperature *T*, in °C, by the equation

$$R = R_0 + kT$$

where $R_0 = 50$ and $k = 0.5$.

The voltmeter displays the value of the voltage across the sensor

$$v_m = \left(\frac{R}{R_s + R}\right) v_s$$

The voltage v_m indicates the temperature, *T*, of the water according to the equation

$$T = \frac{\frac{R_s}{k} v_m}{v_s - v_m} - \frac{R_0}{k} = \frac{150 v_m}{20 - v_m} - 100$$

For example, when $v_m = 11.43$ V, as shown in the figure, the temperature of the water is

$$T = \frac{150(11.43)}{20 - 11.43} - 100 = 100°C$$

Write a C++ program that prints a table showing the meter voltage corresponding to water temperatures varying from 0 °C to 100 °C in increments of 10 degrees.

Submission Instructions

Please follow the instructions below carefully as the submission process consists of two important steps (pay attention to the grading guidelines section):

1) **Submit Printed Hard Copy:**

   By **5:30pm on Tuesday, March 5, 2013**, you must submit a stapled hard copy of following (in order):

   - Coversheet (see below) → First Page
   - C++ Program source code for each problem
   - Grade Sheet (see below) → Last Page

   Please ensure that all the required details are provided on the cover sheet and the grade sheet including (fill in areas where you see red arrow): your name, student ID, lecture section number, and tutorial session number. It is important to include your tutorial session number on the left hand corner on both sheets.

   Submission Location: On the attached coversheet, locate your tutorial session and determine the drop box number. Drop a hard copy of the assignment by the above due date in the appropriate drop box. Drop boxes are located at the first floor in the FEAS Engineering building (ENGR), near the elevators.

2) **Attend Tutorial:**

   Attend your tutorial session (Wednesday Mar 6 / Thursday Mar 7) to demo your solutions to the TA. During the demo, the student will be tested on concepts related to the assignment.

General Guidelines

1) In an effort to help prevent plagiarism, assignment source code will be subject to textual similarity review. If a student submits a source code that is deemed similar in some form to one or more other submissions will automatically result in a zero grade for the assignment.
2) Formatting and indenting your source code is important. Follow the programming style presented during lectures and described in the textbook.
3) If any of your programs does not compile or run, and you are aware that it has some bugs, document this fact at the top of the source code.

Grading Penalties

1) -5% for not ordering sections properly (Not Organized)
2) -5% for not using the cover sheet as first page (No Cover Sheet)
3) -5% for not including grade sheet as final page (No Grade Sheet)
4) -50% for not attending and presenting demo during assigned tutorial date (No Demo)
5) -100% for no printed hard copy of assignment (No Submission)

NO LATE ASSIGNMENTS WILL BE ACCEPTED NO MATTER WHAT IS THE REASON.

You are given more than enough time to finish the assignment well before the due date.

Grading Guidelines

Criteria	Marks	Description
User Friendliness	3.0	Program description message Program output is presented on screen neatly
Program Source Code	4.5	Program is properly documented Meaningful variable names, properly formatted source code Appropriate control structures have been applied
Testing & Correctness	6.0	Programs are complete and implemented correctly
Questions	6.5	Questions related to program source code / implementation
Total	**20 marks**	

References:

- D. Etter, J. Ingber, Engineering Problem Solving with C++, Third Edition, 2011, Pearson.
- C. Horstmann, C++ for Everyone, Second Edition, 2010, John Wiley.
- N. Dale, C. Weems, Programming and Problem Solving Using C++, 2010, Jones and Bartlett

Write your Tutorial # here

ENGR 1200U: Introduction to Programming for Engineers

Assignment 2 Coversheet

Name	
Student ID	
Lecture Section Number (check appropriate section)	☐ 70607 Dr. Q. Mahmoud (Tues & Thurs) ☐ 70610 Dr. E. Al-Masri (Mon & Wed) ☐ 71041 Dr. E. Al-Masri (Tues & Thurs)

Drop Box #	Tutorial #	CRN	Teaching Assistant	Location	Day	Time
13	01	72796	Krupa Kuriakose	UA2230	Wednesdays	1:10 pm – 3:00 pm
13	02	70621	Krupa Kuriakose	UA2140	Thursdays	8:10 am – 10:00 am
13	03	70620	Krupa Kuriakose	UA2230	Thursdays	7:10 pm – 9:00 pm
14	04	72144	Musharaf Rabbani	UA2230	Thursdays	8:10 am – 10:00 am
14	05	72795	Tina Mirfakhraie	UA2220	Wednesdays	1:10 pm – 3:00 pm
14	06	71087	Tina Mirfakhraie	UA2220	Thursdays	9:10 am – 11:00 am
15	08	70617	Titus Okathe	UA2230	Wednesdays	7:10 pm – 9:00 pm
15	07	71159	Titus Okathe	UA2120	Thursdays	9:10 am – 11:00 am
15	09	71160	Saadia Gauhar	UA2240	Thursdays	8:10 am – 10:00 am
16	10	70615	Shukla Shivam	UA2220	Wednesdays	7:10 pm – 9:00 pm
16	11	70622	Shukla Shivam	UA2130	Thursdays	8:10 am – 10:00 am
16	12	70619	Shukla Shivam	UA2220	Thursdays	7:10 pm – 9:00 pm

Write your Tutorial # here

UOIT
CHALLENGE INNOVATE CONNECT

ENGR 1200U: Introduction to Programming for Engineers

Assignment 2 Grade Sheet

Name	
Student ID	
Teaching Assistant (To be completed by TA)	☐ Krupa Kuriakose ☐ Titus Okathe ☐ Musharaf Rabbani ☐ Saadia Gauhar ☐ Tina Mirfakhraie ☐ Shukla Shivam

User Friendliness (3 marks)

1.5 marks: Program description message
1.5 marks: Program output is presented on screen neatly

Program Source Code (4.5 marks)

1.5 marks: Program is properly documented
1.5 marks: Meaningful variable names, properly formatted source code
1.5 marks: Appropriate control structures have been applied

Testing and Correctness (6 marks)

2.0 marks: Speeding car program is complete and implemented correctly
2.0 marks: Credit card number program is complete and implemented correctly
2.0 marks: Meter voltage program is complete and implemented correctly

Questions (6.5 marks)

1.5 marks: Speeding car implementation
2.0 marks: Credit card number implementation
3.0 marks: Meter voltage implementation

Total (20 marks)

Additional Comments (To be completed by TA): → _____ ☐ No Grade

☐	-1	Not Organized
☐	-1	No Cover Sheet
☐	-1	No Grade Sheet
☐	-10	No Demo
☐	-20	No Submission