

Final Exam of Computer Programming 2 (CSW 234)

Level: 2<sup>nd</sup> Level  
Date: 19-01-2019

Academic Year: 2018/2019  
Semester: Fall 2018  
Time: 3 hours

Answer about the following questions

(Total: 60 marks)

**Question 1:**

[15 Marks]

Declare and implement a class ***fraction*** that has two data members: ***num*** and ***denom***, (both of type **int**), which represent the fraction's numerator and denominator, respectively. It has two member functions: ***getFract()*** that reads the ***num*** and ***denom*** of a ***fraction*** object from the keyboard in fractional form (i.e. num/denom), and ***showFract()*** that displays a ***fraction*** object in fractional form. It also has two overloaded operators + and \*, for performing the addition and multiplication operations on any two objects of class ***fraction***, respectively. Note that, the formulas for adding and multiplying two fractions a/b and c/d are as follows:

$$\frac{a}{b} + \frac{c}{d} = \frac{a * d + b * c}{b * d}, \quad \frac{a}{b} \times \frac{c}{d} = \frac{a * c}{b * d}$$

Write a program that declares two objects of class ***fraction***, and repeatedly reads data for these two objects from the keyboard, then perform the two operations on them, displays the results, and asks the user whether he/she wants to continue or not.

**Question 2:**

[15 Marks]

Define a base class **Person** that has two data members: ***name*** (char array) and ***idno*** (int). From this class derive two classes: (1) **Student**, which has an additional data member ***grade*** (char); and (2) **Professor**, which has two additional data members: ***rank*** (char array) and ***nPublications*** (int). Each of the three classes should have two member functions: ***getdata()*** to get its data from the keyboard, and ***showdata()*** to display its data.

Then, write a main program that creates an array of 10 pointers to **Person**. In a loop, gets from the user data about a student or a professor, and use **new** to create an object of type **Student** or **Professor** to hold the data, and then put the pointer to the created object in the array. When the user finishes entering the data for all persons, display the data for all persons entered.

**Question 3:****[15 Marks]**

Define a class *counter*, which has three data members: **count** (of type int) that represents the value of the counter, **overflow** (of type bool) that is set to *true* if an overflow occurs, and **maxcnt** (of type int) that represents the maximum value of the counter. It has a constructor **counter**(int max\_count), that creates a counter with a given maximum count and an initial value 0, and sets *overflow* to *false*, and it has 4 member functions: **increment**(), which increases the current count by 1; but if the current count equals the maximum count, the *count* is set to zero, and *overflow* to *true*; **reset**(), which sets the counter to zero and overflow to false; **get\_count**(), which returns the current count, and **check\_for\_overflow**(), which returns true if an overflow has occurred and false if it has not.

Write a main program that reads a sentence, and counts the frequency of each vowel (a, e, i, o, u) in it, then displays these frequencies. The program should use a separate counter for each vowel, and it should display an error message if any of the counter's overflows.

**Question 4:****[15 Marks]**

A complex number consists of two parts: a real part and an imaginary part, and takes the form: (real\_part + i (imaginary\_part)). Two of the operations that can be performed on any two complex numbers,  $u = a + i b$  and  $v = c + i d$ , are:

Addition:  $u + v = (a+c) + i (b+d)$ , Multiplication:  $u * v = (ac-bd) + i (ad+bc)$ .

Declare and implement a class **Complex** that has: two data members of type float: *real* and *imag*, representing the real and imaginary parts of a complex number, and a **constructor** with no arguments that initializes the real and imaginary parts of an object of this class with zeros. It also has operator functions (+ and \*) to perform the above operations on objects of this class, an **extraction operator** >>, that reads a **Complex** object in the form a+ib, an **insertion operator** <<, that displays a **Complex** object in the same form,

Using this class, write a main program that declares two **Complex** objects, asks the user to enter data for the two objects and an operator (+ or \*), and performs the required operation, then displays the result.

---

*With My Best Wishes*