

PS # 4

Deadline: 01.04.2020 at 23:50

Hand-in Policy:

Create a cpp file for each part of the assignment and save the screenshot while these parts are running.

Your assignment should consist of 4 cpp files and 4 screenshots.

Zip all files of the assignment. Your zip files should look like this

ps4_studentnumber_name_surname.zip

1)

Write a grading program for a class with the following grading policies:

- a. There are two quizzes, each graded on the basis of 10 points.
- b. There is one midterm exam and one final exam, each graded on the basis of 100 points.
- c. The final exam counts for 50% of the grade, the midterm counts for 25%, and the two quizzes together count for a total of 25%. (Do not forget to normalize the quiz scores. They should be converted to a percentage before they are averaged in.)

Any grade of 90 or more is an A, any grade of 80 or more (but less than 90) is a B, any grade of 70 or more (but less than 80) is a C, any grade of 60 or more (but less than 70) is a D, and any grade below 60 is an F. The program will read in the student's scores and output the student's record, which consists of two quiz and two exam scores as well as the student's average numeric score for the entire course and final letter grade. Define and use a structure for the student record.

Data for the test run:

```
1 7 10 90 95
2 9 8 90 80
3 7 8 70 80
4 5 8 50 70
5 4 0 40 35
```

2)

In an ancient land, the beautiful princess Eve had many suitors. She decided on the following procedure to determine which suitor she would marry. First, all of the suitors would be lined up one after the other and assigned numbers. The first suitor would be number 1, the second number 2, and so on up to the last suitor, number n. Starting at the first suitor she would then count three suitors down the line (because of the three letters in her name) and the third suitor would be eliminated from winning her hand and removed from the line. Eve would then continue, counting three more suitors, and

eliminating every third suitor. When she reached the end of the line she would continue counting from the beginning.

For example, if there were six suitors then the elimination process would proceed as follows:
123456 initial list of suitors, start counting from 1

12456 suitor 3 eliminated, continue counting from 4

1245 suitor 6 eliminated, continue counting from 1

125 suitor 4 eliminated, continue counting from 5

15 suitor 2 eliminated, continue counting from 5

1 suitor 5 eliminated, 1 is the lucky winner

Write a program that uses a vector to determine which position you should stand in to marry the princess if there are n suitors. You will find the following function from the Vector class useful:

```
v.erase(iter); // Removes element at position iter
```

For example, to use this function to erase the fourth element from the beginning of a vector variable named theVector , use

theVector.erase(theVector.begin() + 3); The number 3 is used because the first element in the vector is at index position 0.

3)

The type `Point` is a fairly simple data type, but under another name (the template class `pair`) this data type is defined and used in the C++ Standard Template Library, although you need not know anything about the Standard Template Library to do this exercise. Write a definition of a class named `Point` that might be used to store and manipulate the location of a point in the plane. You will need to declare and implement the following member functions:

- a. A member function `set` that sets the private data after an object of this class is created.
- b. A member function to move the point by an amount along the vertical and horizontal directions specified by the first and second arguments.
- c. A member function to rotate the point by 90 degrees clockwise around the origin.
- d. Two `const` inspector functions to retrieve the current coordinates of the point.

Document these functions with appropriate comments. Embed your class in a test program that requests data for several points from the user, creates the points, then exercises the member functions.

4)

Define a class called `Pizza` that has member variables to track the type of pizza (either deep dish, hand tossed, or pan) along with the size (either small, medium, or large) and the number of pepperoni or cheese toppings. You can use constants to represent the type and size. Include mutator and accessor functions for your class. Create a void function, `outputDescription()`, that outputs a textual description

of the pizza object. Also include a function, `computePrice()`, that computes the cost of the pizza and returns it as a double according to the following rules:

Small pizza = \$10 + \$2 per topping

Medium pizza = \$14 + \$2 per topping

Large pizza = \$17 + \$2 per topping

Write a suitable test program that creates and outputs a description and price of various pizza objects.