Homework 7: Structs and Classes

CS16 - Winter 2021

Due:	Thursday, February 25, 2021 (11:59 PM PST)
Points:	50
Name:	_____
Homework buddy:	_____

- You may collaborate on this homework with **at most** one person, an optional "homework buddy."
- **Submission instructions:** All questions are to be written (either by hand or typed) *in the provided spaces* and turned in as a single PDF on Gradescope. If you submit handwritten solutions write legibly. We reserve the right to give 0 points to answers we cannot read. When you submit your answer on Gradescope, **be sure to select which portions of your answer correspond to which problem** and clearly mark on the page itself which problem you are answering. We reserve the right to give 0 points to submissions that fail to do this.

1. (4 points) Write a definition for a structure type for records consisting of a person's wage rate (dollars per hour), accrued vacation (in whole days), and status (hourly or salaried represented as either 'H' or 'S', respectively). Call the type `EmployeeRecord`.

2. (6 points) Given the following structures defined:

```
struct Date {
   int day;
   int month;
   int year;
};

struct Person {
   string name;
   Date dateOfBirth;
};

struct ProjectTeam {
   Person MemberA, MemberB;
   Person Leader;
   string projectName;
   double projectBudget;
   Date projectDueDate;
};
```

If we declare `ProjectTeam TheATeam;` which was then initialized fully and correctly:

a. (2 points) How would you print (to standard out) the project budget for `TheATeam`?

b. (2 points) How would you print (to standard out) the name of Member B of `TheATeam`?

c. (2 points) How would you print (to standard out) the year that the project leader of `TheATeam` was born?

3. (5 points) What's the difference between a `struct` and `class` in C++?

4. (5 points) What's the difference between `public` and `private` members of a class in C++?

5. (5 points) What are class constructors?

6. (25 points) Suppose your program contains the following class definition:

```
class Point {
   public:
      Point(double n1, double n2);
      Point(); // initializes member variables to 0
      double get_x(); // returns value of x
      double get_y(); // returns value of y
      void set_x(double n); // sets a new value for x
      void set_y(double n); // sets a new value for y
   private:
      double x, y;
};
```

a. (12 points) Given the comments shown, give definitions to all 6 of these member functions/constructors:

For points $(x_1, y_1), (x_2, y_2)$, the Euclidean distance formula is given by:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Suppose we want to add a member function to the `Point` class that computes the distance between a given point and itself. Call it `distanceFrom`. The function should take as argument *another* object of type `Point` and return the computed distance. Assume that the `<cmath>` library is already included.

b. (2 points) Give the member function *declaration* for the `distanceFrom` member function.

c. (4 points) Give the member function *definition* for `distanceFrom`.

For a point (x, y), we can rotate it by θ degrees to obtain a new point (x', y'):

$$x' = x\,cos(\theta) - y\,sin(\theta)$$
$$y' = x\,sin(\theta) + y\,cos(\theta)$$

Suppose we want to add a member function to the `Point` class that rotates the point by a given degree and *updates* the values for the member variables `x` and `y`. Call it `rotate`. The function should take as argument a double representing the degree θ. Assume that the `<cmath>` library is already included.

d. (2 points) Give the member function *declaration* for the `rotate` member function.

e. (5 points) Give the member function *definition* for `rotate`.