

CS1337 – Fall 2016

Assignment #4 – Class & Object II

November 7th, 2016

Due Date: December 4th, 2016 – 11:59pm

Specifications: In this assignment, you are going to start with the classes that you developed in the previous assignment.

Before enumerating the changes/additions, let's briefly discuss the purpose of this program. The idea here is that your program will simulate students, initially placed at random position on campus (inside or outside the school building), wandering about the campus grounds picking up trash in an effort to gain rewards. Eventually, he/she/it will enter the school building and, after encountering the teacher, will be assigned a grade for the project based on what they collected. When all classmates have received a grade, a MOST INTELLIGENT STUDENT AWARD will be given and the winner recognized.

Add one more class: **Person Class:**

member variables:

- a string (either type) variable for the person's first name.
- a string (either type) variable for the person's last name.
- an integer variable for the object's intelligence quotient (iq).

member functions: (all public)

- a default **constructor** to which you can assign person's name to "" (empty string) with an IQ of 0.
- Accessors and mutators (functions for member variables).

Add one Structure for Position with two member variables: pos_x and pos_y.

Add one Structure for Dimension with two member variables: width and height.

Modifications to the **Student Class:**

1. Student Class will be a derived class from **Person** class. (you need to modify certain member variables/functions accordingly).
2. **Replace** the place() function with a go_to_campus () function that is passed a campus object by reference. The student object will call its go_to_campus() function, which "places" the classmate object in an empty cell of the campus at random. "Places" means that the chosen cell is given a value of 'S' (for student) and the student's location is set to the cell's location.
3. Each student object will have "pockets" into which he/she/it puts trash objects encountered in their travels about the campus. Implement the "pockets" using

an array of Trash type objects (Trash is described below). Make the max number of items a student can hold 10. Naturally, the student will also need to keep track of how many pieces of trash they have.

4. The move() function will be modified as follows. It will have to have a campus dimension as a reference parameter. The student object will move one step (one cell) per call. If the student steps on a cell that is labeled 'T', they pick up the trash and put it in their pocket (array of trash). So, where does the Trash item come from? Your code will generate a Trash item using its default constructor "on the fly" - meaning it is created at that time the item is picked up; there are no pre-existing trash items. If the student's pockets are full, then the lowest valued item in the pockets is discarded and the new item replaces it (unless, of course, it is the most worthless crap he has picked up so far). Any discarded item just vanishes... maybe it was eaten. There is also a special type of trash – bomb. Any bomb picked up, gets eaten and the student's IQ is diminished by 2 points. (Hence, no bomb ends up in a pocket!) A student can NOT move out of the campus, can NOT move through a school building wall, and can NOT move onto another classmate. You can code the movement of your classmate objects any way you want - random, smart, stupid, searching, whatever. But, once the campus has no more trash, all student objects must move towards and onto the Door of the school building. Once in the building, the student can encounter the teacher and have their project (contents of their pockets) judged and scored. Again, the student's movements in the building can be random or targeted (go for the teacher), and if they step outside again, they will realize there is no trash and must reenter the building. Also, as the student moves about in the building, he/she/it must replace a cell marker with 'S' to show where they are and then replace that cell marker back when they move further. Once a student receives a grade/score, he/she/it is removed from the grid so as to not block access to the teacher by any other student. It is left up to you to determine how smart a classmate's movements are. In summary, your student objects will move about the campus picking up trash until they go in the building; and they must go inside when no more trash exists. And they must move towards the teacher to get their grade in the building.
5. You will need to have the student maintain/keep track whether they are actively moving, what their grade is, etc.
6. The score that a student receives for their project will be the sum of values of the contents.

The Campus Class:

1. It will need accessors and mutators (functions) for the cells of the 2-D array, which we will henceforth refer to as the campus. These functions will naturally need to be passed the location of the cell to be accessed.
2. The campus needs some way to keep track of how much trash is on campus at any time. (This can be done in more than one way, but one way is to have a member variable and a accessor function.)

3. The campus's constructor will randomly place one F (for Teacher) in the school building not in the doorway.

The Trash Class:

- Trash will have a member variable for its name, and a member variable for its value (a positive integer).
- It will have a constructor that will connect to a file of trash items (.txt file) and will pick at random an entry to instantiate the values of the trash item. format is:

name(string) value(int)

The simulation (main):

Your main function will declare a campus object, then two student objects named "John" and "Jane". (You can simulate with more students in your main program). Each student will call their go_to_school() function, placing them at random on campus ground. At this point, your main starts a simulation (a loop) that has each classmate take turns moving until both are assigned a grade/score and a winner is declared. The winner's project must be output for inspection; we don't want any cheaters! Thus, a pseudo-code of your main might look like this:

```
start main
  declare campus
  declare student1, student2
  student1.go_to_school()
  student2.go_to_school()
  display campus
  initialize loop counter
  start simulation - loop on trash gathering
    student1.move()
    student2.move()
    display campus
    increment loop counter
  end simulation - end loop - all classmates scored
  declare/display winner and number of moves to finish
end main
```

Optionals: There are many things you can do with this assignment that can make it even more interesting/funnier/harder/satisfying than it already is. If you have an idea that will add some kind of value to it, go ahead and implement it. Put a big comment at the start of main explaining any additions. Here are some good ideas:

1. include a **speak()** function with the student class. Each time a student object pockets trash, have them say something. For example, Have John say "I found a <name of trash obj>!"
2. Create a teacher class with functionality that makes life interesting for students.

Now, this next information is really important. There is no telling how many iterations your program will go through before your scholars make it into the school building to have their junk piles graded, so the output could be rather large. So, just before you submit, put a conditional statement in your code before the "display campus" code in the simulation that will make it display ONLY THE FIRST 10 LOOPS (showing how both classmates have moved 10 times). After the loop, display it one more time. Your conditional is simply

```
if (counter <= 10)
    cout<<my_school<<endl;
```

As you develop your program, leave the if condition out. That way you can watch what is happening.

The Trash file:

You are welcome to add to the file. Remember to make the first entry in the file the number of items.