

Assignment 07

Link:

<https://docs.google.com/document/d/1J1vx1bZc7Hep6sdo52TLQz1pgUHBWWQ1Naq0l4sCib8/edit?usp=sharing>

Due: Monday, April 18th, 2022, at midnight, 100 points.

Documentation Header Reminder

Before you start your assignment, you will need to add documentation similar to what we demonstrated in the first few lectures.

Function Prototype Documentation Reminder

In the function prototype section of your C++ program, remember to add documentation to each function similar to what was demonstrated in lectures.

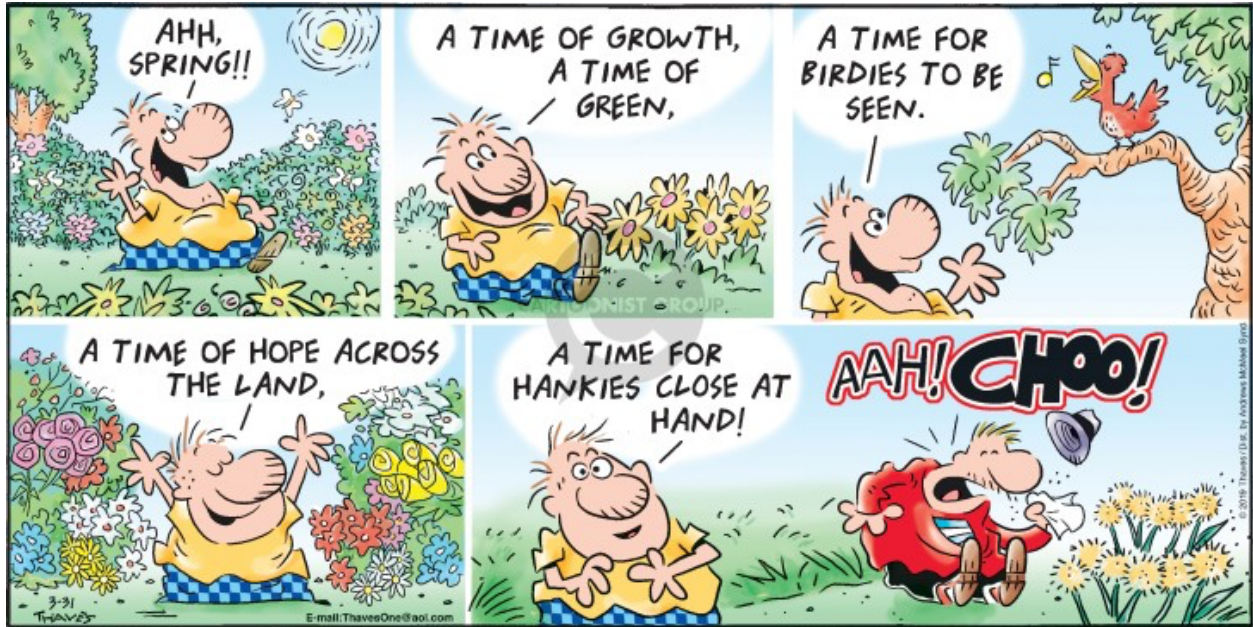
Separate File Compilation

For this assignment (and onward), you will submit multiple C++ compilable files containing a program written in C++. Name your file a meaningful name and give it a proper extension (.h, .hpp, .cpp). Also, make sure that you compile and run your program using the GNU (g++) compiler before submitting to make sure that it will work.

Rock Paper Scissors with Mapmakers

Background

Allergy season has arrived and is ramping up the Kingdom of EET. The king, Rajmund, demanded to have the map of allergies updated as soon as possible. You can't wait to get started as a mapmaker because the one who finishes this task the fastest will receive a great reward!



©Thaves. All rights reserved.

Specifications

For this assignment, you are going to create several classes.

☐ The **Region** class

☐ **Public members:**

☐ A **default constructor** should be made available that does *nothing*. This is simply put in place to guarantee a successful compilation of the program.

☐ A **parameterized constructor** should be made available to take in a region name. It should assign the name to the region's and also initialize the map by calling the following function.

☐ A member function should be made available to **initialize** the allergy data of a region. The three available allergens are animal dander ('A'), dust ('D'), and pollen ('P').

☐ When called, this function should assign a random character value to each grid in the area. The odds that a grid will have animal dander is 21%, dust is 37%, and pollen is 42%.

☐ **Private members:**

☐ A region is represented as an *area* that is troubled with allergies.

- ☐ An area could be structured as a two-dimensional array of characters with fixed array size.
- ☐ A region should have a name.
- ☐ The **Mapmaker** class
 - ☐ **Public members:**
 - ☐ A **default constructor** should be made available to prompt for the mapmaker's name and to initialize the map.
 - ☐ A **parameterized constructor** should be made available to take in a name. It should assign the name to the mapmaker's and also initialize the map properly.
 - ☐ A member function should be made available to **initialize** the mapmaker's uncharted map with an unknown value ('U').
 - ☐ A member function should be made available to **check** if all the unknown areas of the region have been mapped, i.e. no more U characters in the mapmaker's map.
 - ☐ A member function should be made available to allow the mapmaker to **explore** the uncharted area with.
 - ☐ This function should first generate a random *valid* position within the area. If the position is unvisited, then update the map accordingly. If the position has already been visited before, then keep generating a random position until a valid one is found.
 - ☐ **Private members:**
 - ☐ A mapmaker should have a name.
 - ☐ A mapmaker should have a map for the uncharted region.
 - ☐ A mapmaker also needs to keep track of how many times they win Rock Paper Scissors. Set the initialization value to be zero.
- ☐ Additional **global functions**
 - ☐ A global function should be made available to allow the mapmakers to play Rock Paper Scissors with each other. This function should take in two parameters, *mapmaker1* and *mapmaker2*. It will then simulate the game of Rock Paper Scissors between the two players and outputs who's the

winner (or if it's a tie) at the end. The probability of getting a rock, paper, or scissors should be evenly distributed, i.e. one out of three chances. The winning information should also be recorded for the mapmakers too. Most importantly, this function should be a **friend function** to the mapmaker class.

- A global function should be made available to output the **summary** of the program at the end. It should not be a friend function with the mapmaker class.

Overall Program Flow

1. First, create a region called **The Hip Hop Garden** and make sure that its area is properly initialized. The size for the area should be **10** by **4**.
2. Next, create two mapmakers. The first one should be created using the default constructor of the mapmaker class, enter the name **Reimund Ochieng**. The second mapmaker should be created using the parameterized constructor, pass in the name **Pamelia Angelina**.
3. Now, the actual exploration (or competition) begins by having the two mapmakers taking turns. Reimund will go first by performing the following actions in order: (i) explore an unvisited location in the area, (ii) check if the map is complete, (iii) if not complete, then play Rock Paper Scissors with the other mapmaker to motivate your morale. The other mapmaker then takes the next turn and repeats the ordered actions. This counts as one *round* when both mapmakers have performed their course of actions.
4. The main program should end whenever a mapmaker finishes mapping the region. To prevent the mapmakers from becoming too obsessed with playing Rock Paper Scissors with each other, the program should also end when the maximum number of rounds (**388**) is reached.
5. At the end of the program, output a summary of the competition that includes each mapmaker's map, the number of cells explored, and the number of times they won Rock Paper Scissors.

Note

- Set the random seed for this assignment to be **278**.

