# Project 3 of CSCI 13500, Hare and Tortoise Competition, Fall 2020

**Due:** 11 PM Dec 3, 2020. Submit to gradescope.

1. Write a hare and tortoise competition program using object-oriented programming.
2. There is a road with n blocks, where n >= 1. The road can be slippery, so occasionally animals may move backwards from current position.
3. The first animal to reach or pass the last block is the winner; there can also be a tie.
4. In each round, your program adjusts the position of the animals according to its moving pattern. Use variables to keep track of animals’ current positions, which can be at block 0 to n-1.
5. Each animal starts at position 0 (the first block). If an animal slips left before block 0, move that animal back to block 0. You may image there is a road blocker cone at the beginning of the first block, hence animals cannot fall left to block 0.
6. Similarly, if an animal moves past position n-1 (the last block), move that animal back to square n-1. You may image there is a road blocker cone at the end of last block, hence animals cannot fall right to last block.
7. Here are how the hare and tortoise can move typically.

|  |  |  |  |
| --- | --- | --- | --- |
| Animal | Move Type | Percentage of the time | Actual move |
| Tortoise | Fast plod | 50% | 3 squares to the right |
|  | Slip | 20% | 6 squares to the left |
|  | Slow plod | 30% | 1 square to the right |
| Hare | Sleep | 20% | No move at all |
|  | Big hop | 20% | 9 squares to the right |
|  | Big slip | 10% | 12 squares to the left |
|  | Small hop | 30% | 1 square to the right |
|  | Small slip | 20% | 2 squares to the left |

1. Generate the percentages in the above table by producing a random integer in the range 0 <= i < 10. For the tortoise, perform a “fast plod” when 0 <= i < 5, a “slip” when 5 <= i < 7 or a “slow plod” when 7 <= i < 10. Use a similar technique to move the hare.
2. Begin the race.

(8.1) For each round in competition, display a line whose length is the same as number of blocks in a road. Letter T is shown in the position of the tortoise and the letter H in the position of the hare.

(8.2) Occasionally, the animals can land on the same block. In this case, the tortoise bites the hare, and your program should display only letter T, with message "Ouch, Tortoise bites Hare." in the next line.

(8.3) All output positions other than T and H should be blank.

1. After each line is displayed, test for whether either animal has reached or passed the last block. If so, display the winner and terminate the simulation.

(9.1) If the tortoise wins, display

Yay!!! tortoise wins!

(9.2) If the hare wins, display

Yuck, hare wins.

(9.3) If both animal win on the same tick of the clock, you can print out

It is a tie.

## Tasks

1. Design appropriate classes for this project.
2. Use the defined classes to simulate the race.
3. You can change the moving behaviors of the tortoise or hare. You can also change the length of roads to simulate a different race. Here is a sample output.

1 T

Ouch, Tortoise bites Hare.

2 T H

3 T H

4 TH

5 T H

6 T H

7 T H

8 T H

Yuck, hare wins.

Another sample output is as follows.

 is a tie.

1 H T

2 T H

3 TH

4 TH

5 T

Ouch, Tortoise bites Hare.

6 T

Ouch, Tortoise bites Hare.

It is a tie.

1. Key things to learn for this project: class and object design, encapsulation.
2. Write comments, which include but not limit to the author’s name, email, purpose, key ideas and sample output.
3. You need to finish this project by yourself.

## **Important Change when submit**

When you test your code in local computer, you can use srand(time(null)); to randomize the running result when creating a Competition object. However, when you submit to gradescope, you need to use srand(1); Otherwise, gradescope cannot grade a random result.

## Submission

When you need to submit multiple files to gradescope. Do the following.

1. Log in gradescope and click the submission location.
2. In finder app in Mac or file explorer app in Windows, first align files to submit close to each other, then use mouse to draw a rectangle to include all files needed to be upload, names of selected files are highlighted.
3. Drag and paste those files to the submission box of gradescope.

## Compile and run the project in local editor

### Editor vi

1. Put all files (header file ended by .h and C++ source code ended by .cpp) in one directory.
2. To compile all files in this project, enter

g++ \*.cpp

1. Run the project use

./a.out

### Online editor gdb

If you use online editor like [gdb](https://www.onlinegdb.com/) (https://www.onlinegdb.com/),

1. In the original main.cpp provided this editor, comments its contents using multi-line commentor /\* before the first line and \*/ after the last line. Reason: main.cpp contains function main. A project cannot have more than one main function. Note that your project cannot have main.cpp, otherwise there would be name conflict in editor gdb.
2. Upload all your files using by click upload icon .
3. Click Run icon to run the project.