

Problem 1: Bug

Please turn in `bug.h` and `bug.cpp` that contains your implementation of the `Bug` class. I provided starter code `bug.h` and `bug.cpp`. I also provided `bug_main.cpp`, which contains the main function and testers. You should not modify `bug_main.cpp`.

- In `bug.h`, you need to provide the class definition
- In `bug.cpp` you need to provide implementations of the member functions.
- For `bug_main.cpp`, you will not edit this, but you will run this and it will test your implementation.

Please pass all tests before submitting your `bug.h` and `bug.cpp`.

Write a class `Bug` that models a bug moving along a horizontal line. The bug moves either to the right or left. Initially, the bug moves to the right, but it can turn to flip its direction (if the current direction is right, turn to left; if the current direction is left, turn to right). In each move, its position changes by one unit in the current direction.

Provide a constructor

```
Bug(int initial_position)
```

and member functions

```
void turn()  
void move()  
int get_position() const
```

Sample usage:

```
Bug bugsy(10);  
bugsy.move(); // Now the position is 11  
bugsy.turn();  
bugsy.move(); // Now the position is 10
```

- **Note:** There will be no user input. Just run the `bug_main.cpp` and check your answers.

Problem 2: Moth

Please turn in `moth.h` and `moth.cpp` that contains your implementation of the `Moth` class. I provided starter code `moth.h` and `moth.cpp`. I also provided `moth_main.cpp`, which contains the main function and testers. You should not modify `moth_main.cpp`.

- In `moth.h`, you will provide the class definition
- In `moth.cpp`, you will provide the implementations of the member functions.
- For `moth_main.cpp`, you will not edit this, but you will run this and it will test your implementation.

Please pass all tests before submitting your `moth.h` and `moth.cpp`.

Implement a class `Moth` that models a moth flying along the 1D x -axis. The moth has a position, the distance from a fixed origin. When the moth moves toward a point of light, its new position is halfway between its old position and the position of the light source. (If the light position and the moth's current position are the same, then it doesn't move.) Provide a constructor

```
Moth(double initial_position)
```

and member functions

```
void move_to_light(double light_position)  
double get_position() const
```

- **Note:** There will be no output. Please just pass all the tests.

Instructions

- All code must be written originally by yourself. You are not allowed to (even partially) copy code from anyone else. Incident of cheating or plagiarism will be reported to the Dean's office and results in a zero grade in this assignment.
- (5pt) Fill in the template `bug.h`, `bug.cpp`, `moth.h`, `moth.cpp` without changing the file name, and only submit it to Gradescope.
- (5pt) Declare the ownership in the beginning of **each file**. See details in HW1.
- (80pt) Implement all functions correctly.
- (10pt) Write your code with good coding practices, including commenting your code, using descriptive variable names, using constant variables, etc.
- Code compiles with Visual Studio 2022 and solves the questions. Students may lose the majority of points if their code doesn't compile with VS 2022. To receive full credits, the output must look EXACTLY the same as instructed above, including words, spaces, symbols, etc. Your code should not only work for the above examples, but also work for other different inputs.