CMPS 2010 Lab 10

Fall 2022

Part 1 - Preparing New Source Files (Guided)

● Create a new folder called lab10
● Inside of the lab10 folder set up the following files:

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nov 1</td>
<td>05:35</td>
<td>Main.cpp</td>
</tr>
<tr>
<td>Nov 1</td>
<td>05:34</td>
<td>Monster.cpp</td>
</tr>
<tr>
<td>Nov 1</td>
<td>05:35</td>
<td>Monster.h</td>
</tr>
<tr>
<td>Nov 1</td>
<td>05:32</td>
<td>monsters</td>
</tr>
</tbody>
</table>

* (Reminder: you can use the linux command “touch” to create empty text tiles)
  ○ Main.cpp will contain the same code from lab9, except we will be removing the struct Monster{};
  ○ Instead, you will #include “Monster.h” in Main.cpp
  ○ Monster.h will contain the full struct Monster{}; declaration
  ○ Monster.cpp will be blank for now

Part 2 - Convert struct Monster to class Monster

● In Monster.h:
  ○ Change your struct Monster{}; declaration to a class Monster{}; declaration.
  ○ Make sure to include any libraries you will need for this class, like <string>
  ○ Make all of the monster attributes private
  ○ Declare the following public function prototypes:
    ■ Setters: setName, setType, setColor, setEyes, setArms, setLegs
    ■ Getters: getName, getType, getColor, getEyes, getArms, getLegs
● In Monster.cpp:
  ○ Define the getter and setter functions declared in Monster.h
● In Main.cpp:
  ○ Update your code to use the class getter/setter functions rather than accessing the attributes directly. So, for example, instead of updating the name like this:

        monster.name = genName();

  You would do something like this:

        monster.setName(genName());

To compile this weeks labs use: g++ Main.cpp Monster.cpp -o main