ITP 365

Implementing classes and operator overloading

1/26/17

* Struct (structure) allows us to group together related variables into a single variable
* Classes can have both member variables (properties) and member functions
  + AND can prevent other code from directly modifying member variables
  + Private variables – cannot be directly accessed from outside the class
    - **All member variables should be private – known as data encapsulation**
  + Public – can be accessed outside class
  + Getter function
    - Takes no input (return it by value – NOT reference)
    - Ex. int getHours()
      * Return mHours;
  + Setter functions – allow to set member variables
    - Returns void and takes some input
    - Ex. Void setHours (int newHours)
* Splitting up into files
  + Clock.h
    - #Pragma once
    - Maybe #include “string”
  + Clock.cpp – the guts and implementation of the “.h” file
    - #include “Clock.h”
    - #include <iostream>
    - void Clock::reset()
    - **There are 4 parts to every function that belong to a class – return type, class to which it belongs, function name, input**
* Constructors
  + Special type of member function that’s automatically called when an instance of the class is created
  + **Default constructor takes no parameters**
    - Ex. Clock(); -> in “.h”
      * **Function named same as class**
      * **No return type – NOT even void**
    - In “.cpp”
      * Clock::Clock()
        + mHours = 0
        + mMinutes = 0
        + mSeconds = 0
  + **Constructor with parameters**
    - “h.”
      * Clock(int hours, int minutes, int seconds)
    - “.cpp”
      * Clock::Clock(int hours, int minutes, int seconds)
        + mHours = hours
        + mMinutes = minutes
        + mSeconds = seconds
  + Operator Overloading – allows us to define operators for classes/structs we’ve declared
    - friend bool operator==(Clock& left, Clock& right)
      * **friend means that it’s a standalone func. That’s allowed to access private data of the class -> therefore not a class function but since you write friend you have access to private data of class**
      * bool returns boolean
      * operator == -> overloading the operator