CS120 - Computer Science I
Assignment #5
Spring 2014

Due: 5pm Tuesday March 4, 2014.

The purpose of this assignment is to introduce you to classes in a C++ program, and to have you implement some of the *methods* (function members) of a class. The class *interface* (the declaration of the members of the class) has been provided to you, but you will need to finish the implementation of some of the methods of the class.

Our program is a game of keeping the crew of the *Planet Express* happy. This crew (from the show *Futurama*) is represented by the class. It several integer properties, each tied to a specific character:

Thirst	Fry's thirst level
Work	Leela's unfulfilled work orders
Horde	Bender's horde of treasure
Hunger	Zoidberg's hunger level
Beauty	Amy's prettiness
Receipts	Hermes's backlog of receipts to process

Your job is to implement the respective functions for the character's properties. The amounts of each property range from 0-100. The initial values of each property are set at 50 in the constructor (`void Crew::Crew()`) - you can change these if you wish.

`drink()`	Decreases Fry's thirst and increases Leela's work
`deliver()`	Decreases Leela's work and increases Hermes' receipts
`steal()`	Increases Bender's horde and increases Leela's work
`eat()`	Decreases Zoidberg's hunger and increases Leela's work
`primp()`	Increases Amy's beauty and decreases Bender's horde
`account()`	Decreases Hermes' receipts and decreases Bender's horde

Hunger and thirst need to increase a little bit automatically each turn.

You need to implement a scoring mechanism via `int Crew::score()` where the lower the thirst, work, hunger, and reciepts and the higher the horde and beauty, the better the score.

Additionally, the class has the boolean flag *stable* (which is initialized as `true`), representing the crew's ability to continue working together. If at any time one of the crew's respective levels goes out of the specified bounds, this flag needs to be set to `false`, thus ending the game.

You also need to implement the `bool Crew::check()` function, which gets called each round of play. This function needs to return true if the crew is still stable, and false if it is not. The stability corresponds to the score: if the thirst, work, or hunger get too high, or if the horde, beauty, or receipts get too low, the crew is no longer stable, and the Professor's delivery company falls apart (at which point, the game ends). It also needs to print out a warning for each value if it is getting too high (or too low), such as if Hermes has too many receipts (but not enough to crash the company). If it was enough to crash the company (perhaps more than

100), then the function would return false.

This program is an example of a multi-file C++ program. A single file for large programs becomes unwieldy, so instead we separate distinct functionality into distinct files. A typical arrangement is shown here, where the class interface (`crew.h`) and the class implementation (`crew.cpp`) are placed in separate files, with `main.cpp` being in a third file. To compile, the following command can be used:

```
g++ main.cpp crew.cpp
```

Note that the file `crew.h` does not need to be listed, since it is `#included` in the other files.

To turn in this assignment, use `cscheckin` to submit a hw5.zip build from your main.cpp, `crew.cpp` and `crew.h` files with the command

```
zip hw5.zip main.cpp crew.cpp crew.h
```

Also, turn in a printed copy.