IS131

COMPUTER PROGRAMMING — EXERCISES 6

Total Value: 100 points
Due Date: April 14th, 2013 (1800)

For this exercise, we will build a class that stores data about students named Student. The class declaration should be stored in a separate header (student.h) file, and the implementation should be put in a separate (student.cpp) file. Your main() should be in a file of its own (maybe main.cpp?), and properly #include the student.h file. The Student class should have the following private members:

```
      int age;            // Should be from 0 through 130
      string name;        // Should only contain alphabetical characters
      string id;          // Should be in the format A00xxxxxx
                          //(where each x represents a number)
      int grad_year;      // Should be from 1820 through 2099
```

At a minimum, you should provide a default constructor, as well as accessors and mutators for each private member. You may have other constructors if you like.

All functions that attempt to change variables should include exception handling to verify that only legal values (as described in the gray section above) are input. The functions themselves should only throw exceptions, but only your main() should use try and catch, as discussed in class. Avoid terminating the program if you can help it (that is, you shouldn't have an "unhandled exception error") – print a helpful error message and move on. Your main() should exercise each of your functions and should include at least one call to each function that changes variables that intentionally tries to break it. Here's a sample of what you might put in a try block:

```
      Student rook;
      try
      {
        rook.SetAge(500); // He's a vampire
      }
      catch ( /* Something here */ )
      { /* Something here */}
```

In this example, the setAge call should generate an error message after properly handling the exception. Simply adding an if statement and a cout will not earn points!

Submit you source code in a zip, rar, or other archive file via NUoodle. The program will be graded as follows:

75 points: Proper functionality
25 points: Clean code & comments