Lab 05 Specification – Exploring Object-Oriented Languages
Due (via your git repo) no later than 8 a.m., Tuesday, 6th November 2018.
50 points

Lab Goals

Compare basic functionalities of several object-oriented languages. To understand the differences between various object-oriented programming languages. Be able to identify the ways of transforming from one object-oriented language to another.

Four Object-Oriented Languages

In the `lab5` directory of the course repository, locate the three files `Pirate.java`, `Parrot.java`, and `Lab5Demo.java`. Examine them and compile and run them to make sure they work.

As you know from CMPSC 111 [currently CMPSC 100], we often create several Java files, each defining a class, and then combine them into an application. However, we can also place several classes into the same file (not recommended practice, but it's doable)—see file `Lab5OneFile.java`.

The C# language is remarkably similar to Java, at least in the basics. On our system, the C# compiler is named `mcs`. Look at files `Pirate.cs`, `Parrot.cs`, and `Lab5Demo.cs` for the C# versions of the Java programs you just looked at (instructions for compiling are given in the header comments of `Lab5Demo.cs`).

We can also do a "one file" version of the C# program—see file `Lab5OneFile.cs`.

The C++ versions of these programs look significantly different: in addition to the three class files `lab5demo.cpp`, `pirate.cpp`, and `parrot.cpp` we have two header files named `pirate.h` and `parrot.h`. The header files describe the class structure (names of the instance variables and methods), while the bodies of the methods are defined in separate `.cpp` files. Compiling instructions are given in the header comments of `lab5demo.cpp`.

Of course there is an "all-in-one-file" version–see `lab5onefile.cpp`.

Finally, there is the Python version in files `pirate.py`, `parrot.py`, and `lab5demo.py`, as well as the "all-in-one" `lab5onefile.py`.

Look them over, then choose the language (other than Java) you'd like to know more about and study the differences between the Java version and the version in your language of choice.

It would take an enormous amount of space to list all the differences in syntax, etc. between these languages, but I will mention one thing relevant to the assignment below (as an example of what I'm looking for):

> in Python you can't declare multiple constructors. However, you can assign default values to the constructor parameters like this:
>
> ```
> class Thing:
>   def __init__(self,name="anonymous",age = 1):
> ```
>
> Thus, you can write "`x = Thing()`" or "`x = Thing("Fred")` or `x = Thing("Fred",17)`.

Assignment

1. In Java, create two classes:

   - class Gator has two instance variables—a name and a color (they can be strings). It has methods for returning the name and the color and also a method for setting the color. There are two constructors—one specifies both name and color; the other specifies only the name (the color defaults to green). You may create other instance variables and methods at your whim; just remember that I haven't shown you how to do much!
   - class Lab5 tests the Gator class by creating at least two different gators and changing the color of one of them. All information about both gators should be printed out with appropriate labels.

   You may do these EITHER as separate files OR as a single file. I highly recommend attempting the multiple-file implementation.

2. Now choose one of the three languages C#, C++, or Python (you have to pick the one you have the least experience with – I'll be checking your background feedback) and implement a similar program in that language.

   Keep it simple. Here's minimal acceptable output (I hope you will do more); I used python but you might be using mcs or g++:

   ```
   $ python gator.py
   Chompers is green
   Allie is blue
   After calling setColor, Chompers is greenish
   After calling setColor, Allie is blueish
   ```

3. **[Report document required.]** If you were writing a guide for Java programmers explaining to them how to convert their Java classes into classes in one of the languages C++, Python, or C#, choose five things you would mention in such a guide and describe each one of those five things in a separate paragraph. I've already mentioned the one about Python constructors, so you aren't allowed to count that if you chose Python. These can be syntax, or input/output commands, or anything you think a Java programmer would need to modify in order to convert a Java class into a class in your chosen language.

Required Deliverables

Please submit electronic versions of the following deliverables to your Bitbucket repository by the due date:

1. Properly completed and commented source programs.
2. A report document with your guide in PDF format.

Grading Rubric

1. If you complete [Task 1] completely as per the requirement outlined above, you will receive 20 points.
2. If you complete [Task 2] completely as per the requirement outlined above, you will receive 20 points.
3. If you complete [Task 3] completely as per the requirement outlined above, you will receive 10 points (5 points each).
4. If you fail to upload the lab solution file to your git repo by the due date, there will be no points awarded for your submission towards this lab assignment. Late submissions will be accepted based on the late submission policy described in the course syllabus. It is the student's responsibility to communicate to the professor if it is a late submission. If the student had not communicated in advance about the late submission, the lab work shall not be graded as such.
5. Partial credit will be awarded, based on the work demonstrated in the lab submission file.
6. If you needed any clarification on your lab grade, talk to the Professor. The lab grade may be changed if deemed appropriate.