Homework 06, CPSC-4175

Chapter 15, Object-Oriented and Classical Software Engineering

September 25, 2017

1. Consider the programing languages you may have studied and their differences. Think of two problem domains you would use for each language. For each problem domain, explain why your choice of the other language would be a poorer choice for an implementation language.

2. Give some concrete counter-examples of good programming practices as discussed in the book. These include meaningful variable names, self documenting code, symbolic constants, and code layout. In other words, give examples of what *not* to do.

3. Give two versions of coding standards for subprocedure blocks. This may require an internet search. Which do you prefer, and why?

4. An informal description of the DRY principle is this: "The *second* time you write exactly the same code, stop what you are doing and place that functionality in a subprocedure. Then, invoke the subprocedure whenever you need that functionality." Give one example of this from your previous programming experience. If you have never experienced this in practice, make up an example.

5. What is a *stub*? What is a *driver*? Have you ever used stubs or drivers in your previous programming experience? If so, give an example. If not, consider a function that takes a unit price, a sales tax rate, and the quantity ordered, and write a stub (using pseudo code or a language of your choice) that takes these three arguments and returns a value.

6. List two strengths and one weakness of top down integration.

7. List two strengths and one weakness of bottom up integration.

8. This is not in the book and will require some independent research. A *domain specific language* (DSL) is a small, incomplete language typically use as a "glue" language between a lower level language (like C) and a higher level language (like Python). An example of a DSL might be a helper language for HTML, e.g. `div(...)` might resolve to `<div> ...</div>`. How might a DSL promote what the book calls *sandwich integration*? This is not a trick question but it will require some thought. If you work as a developer, you will eventually write your own DSLs for various purposes.

9. What is the difference in testing carried out by the implementation group and the testing carried out by the SQA group?

10. Read the letter by Edgar Dijkstra to the *Communications of the ACM* (in the PDF directory as `dijkstra68.pdf`) and write a one paragraph appreciation of this letter.