Curriculum

Evan Drake

March 19, 2024

Contents

1. Concepts 3
   1.1. Software life-cycle 3
   1.2. Procedural vs OOP 3
   1.3. Programming design tools & programming environments 3
   1.4. Documentation 3
   1.5. Coding conventions 3
   1.6. Data Types, Variables, Expressions, Sequential processing 3
   1.7. Collections 3
   1.8. Control flow 3
   1.9. Simple Algorithms 3
   1.10. File IO 3
   1.11. Error handling 4
   1.12. Passing parameters by value & by reference 4
   1.13. Principles of testing & designing test data 4
2. Lectures (12 weeks) 5
   2.1. Programming design tools 5
   2.2. 5

1. Concepts

1.1. Software life-cycle

- design: waterfall, agile, pseudocode, wireframing
- development: coding with in teams
- styles: style guide
- documentation: in-line comments, block comments, outside docks
- testing: internal / external
- maintenance: ticketing, bugs, QA

1.2. Procedural vs OOP

- Procedural: systems of purely sequential instructions
- OOP: systems designed around objects, large structures describing traits rather than behavior

1.3. Programming design tools & programming environments

- design tools at this stage should include: pseudocode, wire framing, UML's
- environments like what dependant software is necessary to compile code, testing, literal code writing.

1.4. Documentation

- code comments: inline / block
- external docs like a wiki, or manual paper

1.5. Coding conventions

- conventions include: style guide

1.6. Data Types, Variables, Expressions, Sequential processing

- primitive types are: Numbers, Characters, Boolean
- Numbers can be expanded
- RAM, the heap. explain why types and variables are designed the way they are for machines / then maybe in language.

1.7. Collections

- Most primitive of which is an array type

1.8. Control flow

- if then else
- simple iterative loops

1.9. Simple Algorithms

- Sorting: simple naive pivot sort
- Search: simple searching. Explain that it is more efficient once sorted in linear structures

1.10. File IO

- **instream** / **outstream**, using **fstream** package
- sequential access

1.11. Error handling

- simple throw/catch ? **Might be a good idea to use control flow to introduce this**

1.12. Passing parameters by value & by reference

- memory explanation of copy or reference value

1.13. Principles of testing & designing test data

- unit tests
- integration tests
- divide and conquer strategies

2. Lectures (12 weeks)

2.1. Programming design tools

- what is Programming
- pseudoCode
- enviornments
- Design strategies

2.2.