Unit 1 Module 4: Programming Concepts

Dylan Lane McDonald

CNM STEMulus Center
Web Development with PHP

August 12, 2014

Outline

Who Are We?

What are we training to be?

- Software Engineer: one who writes software

Who Are We?

What are we training to be?

- Software Engineer: one who writes software
- Computer Scientist: one who uses sound mathematical principles to solve problems

Who Are We?

What are we training to be?

- Software Engineer: one who writes software
- Computer Scientist: one who uses sound mathematical principles to solve problems
- Software Architect: one who designs software programs so software engineers can implement them

Who Are We?

What are we training to be?

- Software Engineer: one who writes software
- Computer Scientist: one who uses sound mathematical principles to solve problems
- Software Architect: one who designs software programs so software engineers can implement them
- Web Developer: one who writes software for the web programming paradigm

Who Are We?

What are we training to be?

- Software Engineer: one who writes software
- Computer Scientist: one who uses sound mathematical principles to solve problems
- Software Architect: one who designs software programs so software engineers can implement them
- Web Developer: one who writes software for the web programming paradigm

So who are we. . . ?

Who Are We?

What are we training to be?

- Software Engineer: one who writes software
- Computer Scientist: one who uses sound mathematical principles to solve problems
- Software Architect: one who designs software programs so software engineers can implement them
- Web Developer: one who writes software for the web programming paradigm

So who are we. . . ?
All four of them!

What Are We Doing?

As mentioned in the last slide, we are learning the web programming paradigm.

What Are We Doing?

As mentioned in the last slide, we are learning the web programming paradigm.

Definition

A **paradigm** *is how a programming language is structured and thought about. Each paradigm is specialized to the problems it aims to solve.*

What Are We Doing?

As mentioned in the last slide, we are learning the web programming paradigm.

Definition

A **paradigm** *is how a programming language is structured and thought about. Each paradigm is specialized to the problems it aims to solve.*

There are five major paradigms and dozens of minor paradigms. Each paradigm is specialized to a class of problems. Problems can be solved by multiple paradigms and there are advantages and disadvantages of each.

Programming Paradigms

Common paradigms include:

- Imperative: programming step-by-step using procedures (C)

Programming Paradigms

Common paradigms include:

- Imperative: programming step-by-step using procedures (C)
- Functional: programming using mathematical functions (Haskell)

Programming Paradigms

Common paradigms include:

- Imperative: programming step-by-step using procedures (C)
- Functional: programming using mathematical functions (Haskell)
- Object-oriented: programming by modeling each player in the program (Java)

Programming Paradigms

Common paradigms include:

- Imperative: programming step-by-step using procedures (C)
- Functional: programming using mathematical functions (Haskell)
- Object-oriented: programming by modeling each player in the program (Java)
- Logic: programming by axiomatic logical statements (Prolog)

Programming Paradigms

Common paradigms include:

- Imperative: programming step-by-step using procedures (C)
- Functional: programming using mathematical functions (Haskell)
- Object-oriented: programming by modeling each player in the program (Java)
- Logic: programming by axiomatic logical statements (Prolog)
- Symbolic: programming by manipulating formulas & symbols (LISP)

How Do We Use It?

Programming languages are used in three different ways:

- Compiled: Source code is passed to a *compiler*, which takes the source code and transforms in into *machine code*, which is directly executable by the CPU (C)
- Interpreted: Source code is passed to an *interpreter*, which executes the source code line-by-line as the code is encountered (PHP)
- Hybrid: Source code is compiled into an intermediate, optimized format called *byte code* that is saved and passed to an interpreter when it needs to execute (Java)

The other languages we will be covering, JavaScript & SQL, are also interpreted.

How Do We Write Better Code?

Code maintainability is essential. Non readable, non maintainable code precludes development and is a hindrance to the team. Code can be made easy-to-read by:

- Properly indenting the code at each logical level
- Using variable names that concisely describe what the variable does
- Sticking to one variable capitalization convention[1]
- Thoroughly commenting code as you write it
- Exercising good `git` etiquette...

[1] In this class, "camelBackVariables" are preferred.

How Do We Write Better Code?

Code maintainability is essential. Non readable, non maintainable code precludes development and is a hindrance to the team. Code can be made easy-to-read by:

- Properly indenting the code at each logical level
- Using variable names that concisely describe what the variable does
- Sticking to one variable capitalization convention[1]
- Thoroughly commenting code as you write it
- Exercising good `git` etiquette...
  - Commit early, commit often!
  - Write descriptive comments in the `git commit` journal

[1] In this class, "camelBackVariables" are preferred.

How Should I Act?

Attitude is everything! Computer Science is a challenging field in which you are part of a team. Separate yourself by:

- **TENACITY**: Things **will** go wrong at some point. Great programmers persist, keep their frustrations in check, and do what it takes to climb the inevitable brick walls.
- **Teamwork**: Cooperation and being approachable is a must. You will be working side-by-side with people and the more you help the team, the more you help yourself.
- **Empathy**: Again, this can be a frustrating field at times. Understanding and taking the point of view of an upset end-user will not only defuse a volatile situation, but also help you arrive at a solution.