

Exam #2 Review

Thomas Barrasso
CS-E1- Fall 2011

Agenda

- + Course notes & Exam #2 details.
- + Introduction to computer programming.
- + Review of website development.
- + Sample questions.
- + Q&A (feel free to raise your hand and ask a question at any point throughout the review).

Course Notes

- + Homework assignment #5 is due by noon of Monday, December 12th 2011.
- + The Final Project is due by noon of Monday, December 19th 2011.
- + Exam #2 will be held on Tuesday, December 6th 2011 in Harvard Hall 201.
- + *Closed-book*, so no notes, textbooks, computers, tablets, smart phones, etc.

Exam #2

- + The 2006 Exam #2 is available for practice, as well as the answer key.
- + It will last **exactly** *2 hours (120 minutes)*.
- + Content will include material mostly from *Lectures 5 - 9* including multimedia, security, web development, and programming but *may* include also content from *Lectures 1 - 4*.



Source Code

- + What a developer/ programmer writes.
- + Text that gets compiled using a *compiler* into *machine code*, or in the case of a *scripting language* interpreted dynamically.
- + *Open-source* is providing your source code to the world under a license like the [GNU GPL](#) such that other programmers can contribute and modify the programming to their liking.

Languages

- + Just as there are numerous languages which one can speak, there are numerous *programming languages* that a computer can understand.
- + Common programming languages include C, C++, C#, Java, JavaScript, Objective-C, PHP, and **many** more.
- + Used to create programs that control the behavior of a machine.

Comments

- + Part of the source code used to inform programmers of the function/ implementation details.
- + Comments are not visible in a compiled program.
- + Important especially when source code is shared.
- + Often denoted with double-forward slashes `//` or `/** */` for multi-line comments.

Variables

- + Much like in math, you give a variable a name and value; i.e., $x = 2$
- + Some programming languages like C are *strictly-typed*, meaning that instead of saying " $x = 2$ " you would say "`int x = 2`" specifying the type.
- + Some programming languages are *dynamically-typed*, meaning no required variables types.

Conditionals

- + IF..THEN..ELSE statements control the *flow* of a computer program.
- + IF a condition is true (using boolean comparison) THEN do something, ELSE do something else.
- + Some programming languages support ELSE IF, which is like ELSE but with an additional clause.

Syntax

```
int x = 2
```

```
// predicate
```

```
IF (x == 2)
```

```
THEN
```

```
// consequence
```

```
ELSE IF (x == 3)
```

```
// another consequence
```

```
ELSE
```

```
// alternative
```

Loops

- + Used to do something more than once.
- + Two common types: FOR and WHILE.
- + Operate by doing something repeatedly, checking a boolean statement much like IF, except looping stops when the statement is no longer true.

Syntax

```
FOR (i = 0; i < 10; i++)  
// do something 10 times
```

```
boolean check = true  
int i = 0  
WHILE (i < 10)  
    // do something 10 times  
    i++
```

Website Development

- + Web pages are created in HTML (Hyper Text Markup Language) which is not a programming language.
- + HTML is composed of a hierarchy of *tags* that each represent a particular function.
- + HTML is combined with CSS (Cascading StyleSheets) to design a web page.
- + HTML documents are served via HTTP (Hyper Text Transport Protocol) from a server to a web browser.

Hierarchy

```
<html>
  <head>
    <title>This is the title!</title>
  </head>
  <body>
    <p>This is a paragraph.</p>
  </body>
</html>
```

Notice the structure? Inside the `html` tag there is a `body` tag, and inside that there is a `p` tag? It forms a hierarchy.

Name That Tag

<p> </p>

<a>

What's Odd?

```
<a href="http://bing.com">Google</a>
```

```

```

```
<a href="http://badguy.com" style="color:  
white;background:white;">visible text</a>
```

CSS

- + Property-value pairs used to denote aspects of a web page's style including color, size, position, etc.
- + Applied inline to an element with the style attribute, or as a separate file using the link tag.
- + When externalized, elements are referenced by their id, class, tag, etc. (either tag name or attribute values).

Inline: `<p style="color:red;">lorem ipsum</p>`

External: `p { color:red; }`

Questions?

fin