RIT and Webster Central School District

CS Principles Pilot

Outline

- Brief introduction to CS Principles
- Discussion of RIT/Webster Pilot
- Exercises that worked
  - Data Representation
  - Stenography
  - Computer Hardware
- Exercises that did not work
  - Intro Programming/Web
  - Accessible Web Design
- Summary – Next Steps
This project would not have been possible with support from Microsoft

Thank you!!!

A New First Course in Computing Computer Science

Principles is a new course under development that seeks to broaden participation in computing and computer science. Development is being led by a team of computer science educators organized by the College Board and the National Science Foundation.

Key players: NSF and College Board
The Course

- **AP Computer Science: Principles** is designed to introduce students to the central ideas of computing and computer science, to instill ideas and practices of computational thinking, and to have students engage in activities that show how computing and computer science change the world.

- The proposed course is rigorous and rich in computational content, includes computational and critical thinking and skills, and engages students in the creative aspects of the field. Through both its content and pedagogy, this course aims to appeal to a broad audience.

Resources

- College Board Principles Web Site

- AP CS: Principles Curriculum Framework
    - Course Annotations
    - Big Ideas
    - Computational Thinking Practices
    - Claims and Evidence

- CS Principles Web Site
  - http://www.csprinciples.org/
Pilots

- Five pilot sites were selected to teach the course during the next phase of the project, including:
  - Metropolitan State College of Denver (Jody Paul)
  - University of California at Berkeley (Dan Garcia)
  - University of California at San Diego (Beth Simon)
  - University of North Carolina, Charlotte (Tiffany Barnes)
  - University of Washington (Larry Snyder)

Additional Resources

- The pilots have resources on the web
  - http://inst.eecs.berkeley.edu/~cs10/sp11/
  - http://www.livinginacomputingworld.org/
  - http://www.cs.washington.edu/education/courses/cse120/11wi/
  - http://csprinciples.cs.washington.edu/blog/
Where We Fit In

- In the fall of 2010 RIT did an *unofficial* pilot of CS Principles
  - Curriculum based on information from the CS principles site
  - Used C# for programming
  - Supported by Microsoft

- Intended Audience
  - CS Majors who do not feel ready for CS1
  - Non-majors
  - Undecided (exploration) students

- Some additional details...
  - A couple of programs decided to require the course
  - *Eventually* can be used for general education and writing intensive credit

RIT Principles Course

- 10 week (quarter-based) course
  - 2 classes each week
    - Lecture
    - Hands on Laboratory Exercise
  - Qualifies for RIT Writing Intensive and General Education Credit

- Assessments
  - Weekly Quizzes
  - Weekly Labs
  - Term Paper
  - Poster
Course Content

- Content is divided into 4 major areas/themes
  - Nuts and Bolts
  - Algorithmic Thinking
  - Computing Systems
  - People and Computing

Nuts and Bolts

- Data Representation (Claims 8, 9)
  - Binary and Hexadecimal
  - Characters
  - Video/Images (Color representation)
  - Music

- Hardware (Claims 24 and 25)
  - Basic Computer Organization
  - CPUs
  - Memory
  - Video Cards/GPUs
  - Input/Output
  - Storage

- System Software (Claims 5, 24 and 25)
  - Operating systems
  - Operating system updates and patches
  - Antivirus software
  - Administrative tools
Algorithmic Thinking

- Problem Solving (Claim 5, 11, 12, 13 and 19)
  - Algorithms and Algorithmic Thinking
  - Basic constructs – conditionals, loops, variables
  - Developing algorithms

- Software Development (Claims 2, 11, 13, 16, 18, 20 and 23)
  - The Software Development Process
  - Programming Languages
  - Collaborative Programming
  - Introduction to Programming in C#

- The Limits of Computing (Claims 6, 14, 15, 27)
  - Can Everything be Computed?
  - Hard and Really hard Problems
  - Basic Time/Space Complexity

Computing Systems

- Networking (Claims 1, 10, and 21)
  - How it all Works
  - Hardware that makes it Happen
  - The Internet
  - Keeping your information from prying eyes
  - Social Computing

- Knowledge is Power (Claim 7, 28, 29 and 30)
  - Databases
  - Data Mining
  - Search Engines: The Good, the Bad, the Scary
  - Data is not Information

- Computing systems (Claim 22)
  - Power in numbers
  - Scalability
  - Distributed and cloud computing
  - Parallel computing
People and Computing

- Ethical Computing (Claim 3, 4, 10 and 26)
  - Good practices
  - Information access and sharing
  - Electronic stalking vs. information gathering
  - Safe Computing Practices
    - Viruses/Worms
    - Malware
    - Phishing
    - Protecting your personal information
    - Social engineering
    - Living off the Grid

- Accessibility (Claim 17)
  - Networks
  - Devices
  - Interfaces
  - Human factors
  - User interfaces

RIT Pilot Phase II

- Continuing the Pilot in fall 2011
  - Course will be taught at RIT and Webster
    - 2 sections (quarter-long) at RIT
    - 2 sections (year-long) at Webster

- Being co-developed by
  - Trudy Howles (RIT)
  - Sage Miller (WCSD)
  - Paul Tymann (RIT)

- Webster students will be able to request to take exam to receive dual-credit for the course
Labs that Worked

- Image Manipulation
  - Turn a picture to gray scale

- Steganography
  - Learn how to hide a picture in a picture

- Go Buy a Computer
  - Given some requirements build a machine

Image Manipulation

- Lecture content for the week data representation
  - Included images in the discussion

- Lab
  - Complete a program that will convert a color image to gray scale
The Code

```java
int average = // add code to compute the average of the three color values
if // add a boolean expression that is true if the average is less than 100
{
    p.Red = 220;
    p.Blue = 220;
    p.Green = 220;
}
else if (average >= 100 && average < 200)
{
    p.Red = 169;
    p.Blue = 169;
    p.Green = 169;
}
else // The average must be >= 200
{
    p.Red = 105;
    p.Blue = 105;
    p.Green = 105;
}
```

Picture Before Manipulation
Picture After Manipulation

What They Did!!
Ideas For The Future

- Gray scale is not what is important – the idea that you are manipulating the image is

- So maybe...
  - Colorization?
  - Color swapping/erasing
  - Green screen

Steganography

- Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message.

- Lab
  - Learn how insensitive our eyes are to 255 colors
  - Complete a program that will extract an image hidden in another

- This lab was inspired by a similar lab done by Owen Astrachan and Larry Snyder
Mask Colors

Power of the MSB
Steganography in Action

Steganography in Action
### The Code To Do It

```c
for (int i=0; i<srcw; i++)
{
    for(int j=0; j<srch; j++)
    {
        c = get(i,j);
        if (i<wid && j<hi)
        {
            cprime = color(64*(int(red(c))%4),
                            64*(int(green(c))%4),
                            64*(int(blue(c))%4));
            set(i,j, cprime);
        }
        else
        {
            set(i,j,c);
        }
    }
}
```

---

### Ideas For The Future

- Probably no where nearly as exciting...
  - Embed an ASCII message in an image using only the LSB
  - Could then show how nearly identical the images are
- Perhaps discuss digital watermarks
Go Buy a Computer

- Lecture content for the week computer organization and hardware
  - Discussed various components and all of the abbreviations

- Lab
  - Given a requirement for a computer – find a machine that fits those requirements
  - Present your machine to the class and explain your decisions

Sample Requirements

- Computer required for email, surfing the web, word processing. Maybe store some photos and music

- Computer required primarily to play games

- Computer needed for someone who travels. Will be used to access email and internet while on the road

- Computer required for college student. General email and internet usage. Word processing and power point. Some programming
Labs that Bombed

- Generate a Simple Web Page
  - Complete a program that generates a web page

- Accessibility For All
  - Find web sites that are and are not accessible

What Were We Thinking?

- I am not sure what we were thinking when we developed this lab
  - Might have been worried about finding time to cover all topics

- Students were totally confused
  - Some thought you needed to write a program to generate HTML
  - Others could not discern “programming” from developing a web page

- Obviously this lab needs to be broken into two parts
  - Generate a web page
  - Work on a simple C# program
Lecture content for the week included basic accessibility issues

Lab
- Spend 40 minutes surfing the web trying to find at least one web site you feel does a good job accommodating all users, and one site that fails to do so.
- Prepare a list of what you think the site designers did well (or poorly). In the second hour, you and your partner will present a quick demo of the 2 sites you found and discuss the following:
  - The "good" site: what did the site designers do particularly well?
  - The "weak" site: describe 1-2 things you would propose to help improve the accessibility of the site.

The lab was really not a disaster, students did not do what was expected
- Most focused on the Interface Hall of Shame ([http://homepage.mac.com/bradster/iarchitect/shame.htm](http://homepage.mac.com/bradster/iarchitect/shame.htm))
- Others found sites dedicated to showcasing “ugly/bad” web designs

The discussion did not cover much about accessibility but students did talk about web design
- Might be useful to re-work and use it for this purpose. Students did enjoy the lab

Probably part of the problem is that most students do not really understand the “need” to make something accessible
Summary – Next Steps

- Over the next year the RIT course will be refined
  - Better lab assignments
  - Develop resources for both college and high school teachers
  - Get an understanding of the difference between teaching the course in college versus high school

- Phase III
  - Offer more extensive workshop
  - Roll the course out to more schools in the area