

CPR 101

Welcome to Computer Skills for Programmers

Agenda



Discussion



- Course Format and Evaluation
- About Programming and Scripting
- History of Programming
- Styles and Methodologies
- Levels



Goal of CPR101

- Make you a better programmer
 - Learn how applications and system software interact
 - Modern network concepts, cloud computing and virtualization
 - Numbering systems, version control and project management,

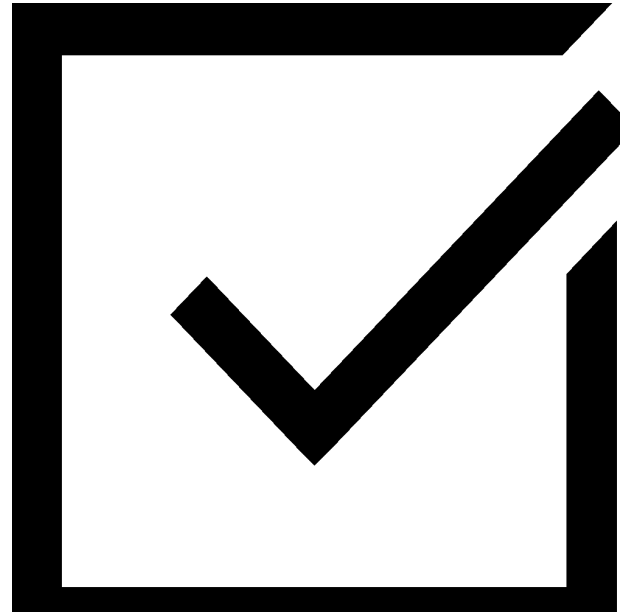


Course Format Evaluation & Success



Quiz

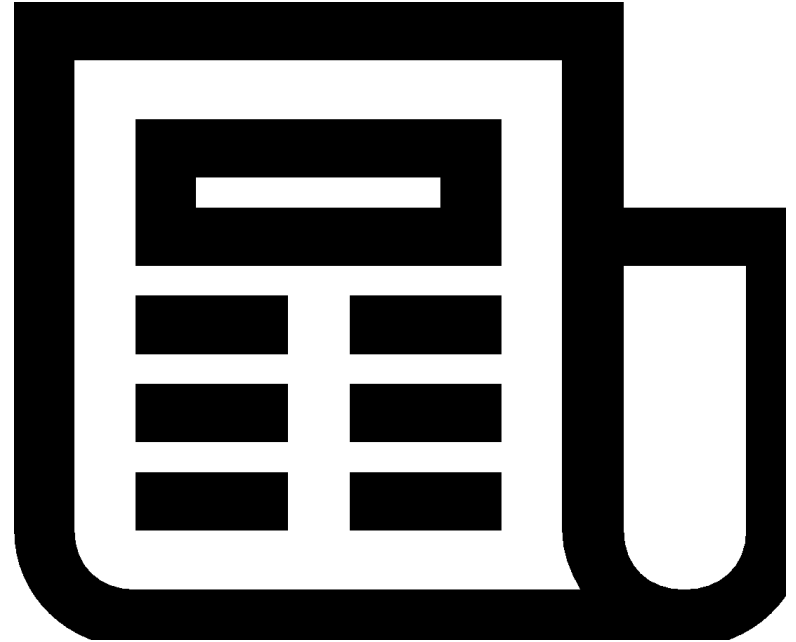
- Each class will begin with a 15 minute quiz (5 short answer questions)
- You will write 10 quizzes, the best 8 will be selected for grading
- If you are absent for a quiz, a mark of “0” will be assigned



News of the Week

i

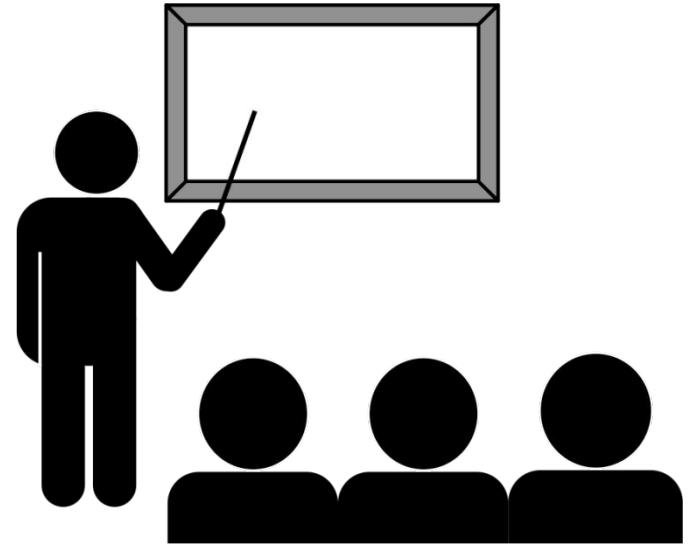
- After the quiz there will be a 15 minute discussion on a current news topic
- You are responsible for subscribing to a news service (Activity 1- Part 1)
- Come prepared to participate
 - Build vocabulary
 - Expand knowledge beyond the classroom



Weekly Lecture

i

- Lectures are 80 minutes in length and are instructor centred
- Ask questions anytime during the lecture
- Be on time
- Silence cell phones
- Be respectful of others



Weekly Activity

i

- Weekly Activity is a 50 minute exercise for skill development and knowledge to complement the weekly lecture material
- Graded on completing activity and reflection notes of what you learned
- Must complete 9 out of 11 activities to pass the course
- Activities must be completed in class to receive a grade. Missed activities will receive a grade of “0”



Final Assignment

- The Assignment must be completed to pass the course
- The assignment will be worked on in teams and is worth 15%
- You will have in class time to complete it (last 2 weeks of class)



Graded Work

Content	%
Quizzes (8 of 10)	40%
Activities (9 of 11)	45%
Final Assignment	15%
Total	100%

Standards:

85% Attendance Record for the term (miss 4)
50% Average for all Graded Work



For Success ...

10 Steps For Success ...

1. Review the material each week before class
2. Take notes on course material, including solutions to any problems that arise – The first step in learning
3. Follow technology news and come prepared to participate
4. Thoroughly read all steps in the activity before starting
5. Ask other students or teacher for help when needed
6. Attend all classes and be on time
7. Silence audible devices
8. Focus on the task at hand – don't multitask
9. Review previous lecture notes to prepare for quiz
10. Submit all graded work on time

Programming or Scripting

What is Programming or Scripting?

i

What is Programming or Scripting?

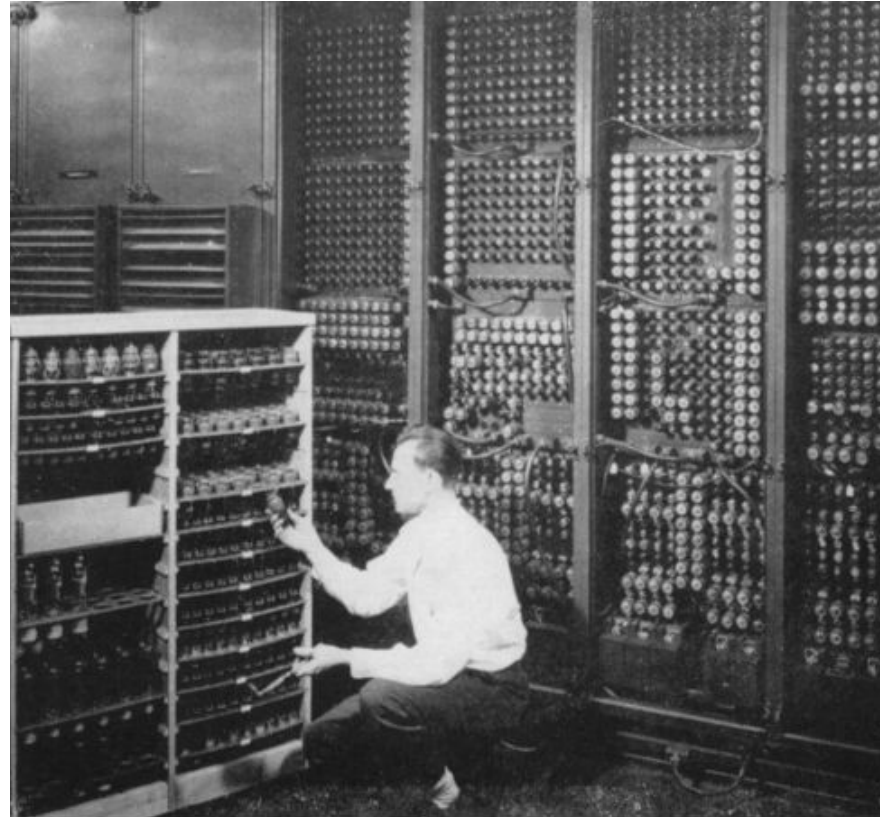
- English like commands
- Written in a specific language
- Contained in a text file called a program
- Give instruction to a computer
- Compiled into machine code
- Executed by CPU



```
252 document.getElementById( 'bigimageDesc' ).innerHTML = description;
253 }
254
255 function updatePhotoDescription() {
256     if (descriptions.length > (page * 9) + (currentImage.substring(0, 1))) {
257         document.getElementById( 'bigimageDesc' ).innerHTML = description;
258     }
259 }
260
261 function updateAllImages() {
262     var i = 1;
263     while (i < 10) {
264         var elementId = 'foto' + i;
265         var elementIdBig = 'bigimage' + i;
266         if (page * 9 + i - 1 < photos.length) {
267             document.getElementById( elementId ).src = 'images/';
268             document.getElementById( elementIdBig ).src = 'images/';
269         } else {
270             document.getElementById( elementId ).src = '';
271         }
272     }
273 }
```

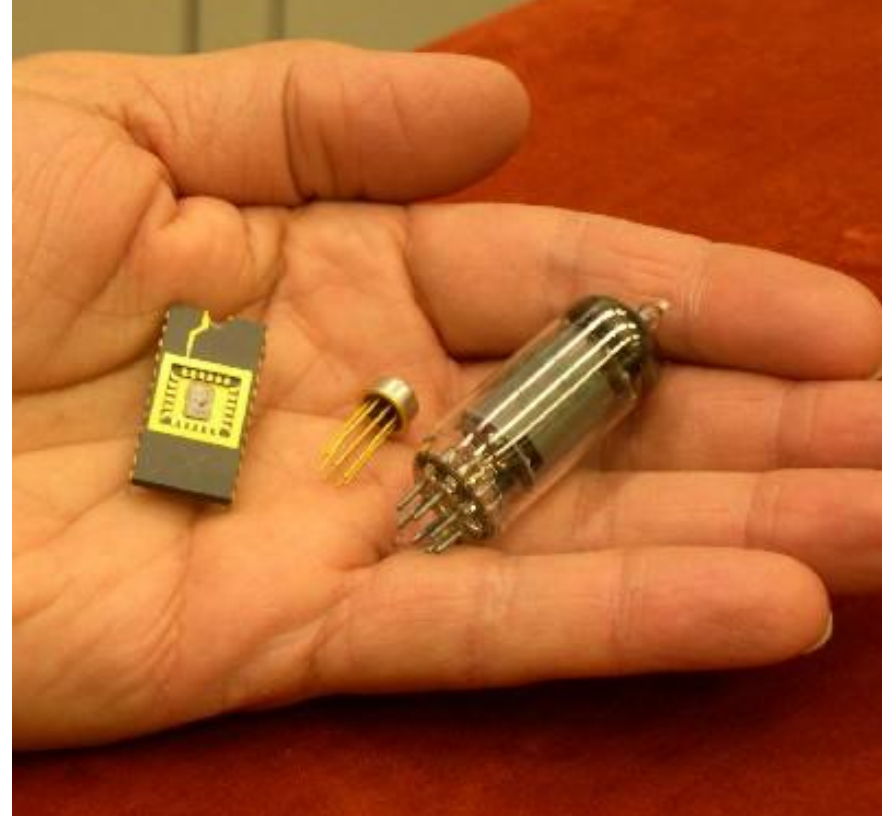
Early computers

- First computing devices in 1800s and early 1900s were more mechanical, very basic computing.
- 1940s saw the use of vacuum-tube computers, room-sized. Each vacuum tube can effectively perform one binary operation.



Transistors

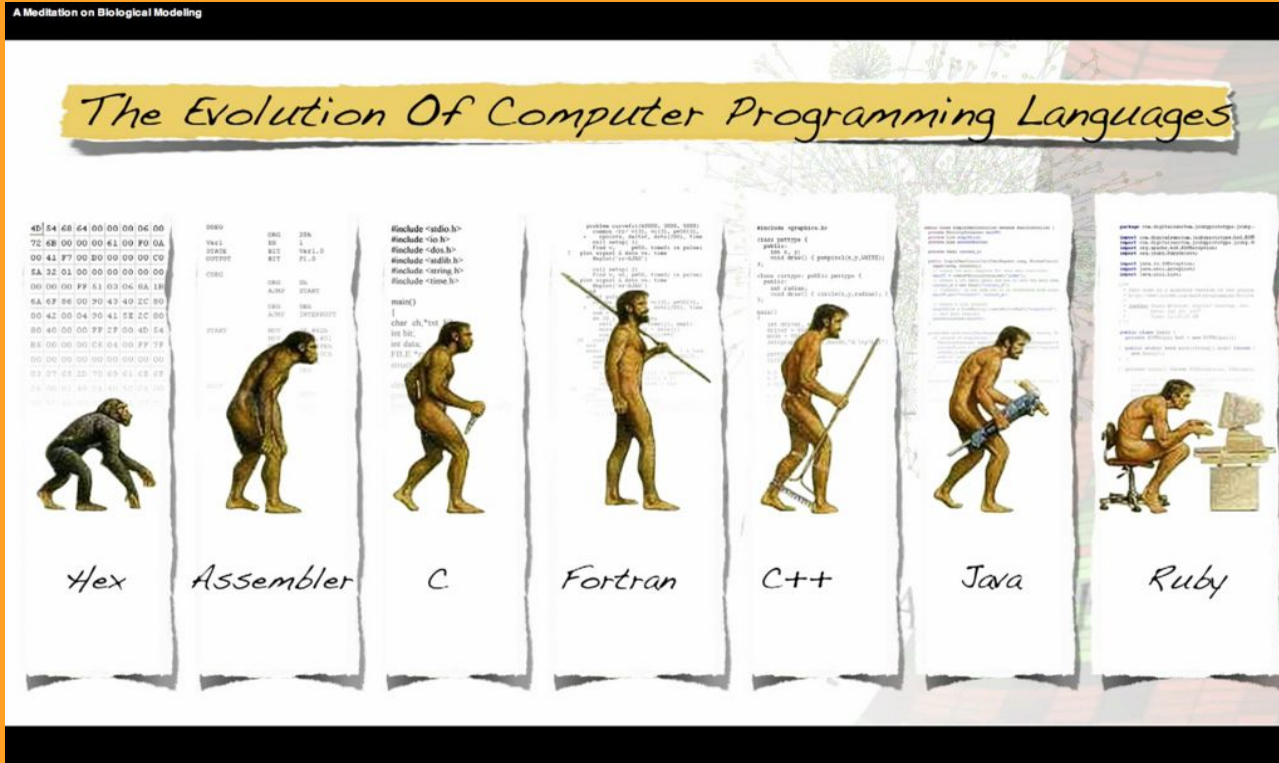
- Transistor invention in 1940s led to microchip, allowed design of smaller computers that could perform more complex tasks
- Smaller, computers allowed for more programming code and what we think of as modern computing devices



“It would appear that we have reached the limits of what it is possible to achieve with computer technology, although one should be careful with such statements, as they tend to sound pretty silly in five years.”

John von Neumann, 1949

Short History of Programming Languages



Programming Languages

i

- Do you know of any programming languages?

1945-1959

- FORTRAN (1945) First widely used high-level language
- LISP (1958), COBOL (1959)

1960-1979

- C (1972) Early systems programming language
- SQL (1978) Query Language

1980-1989

- C++ (1980) Still used heavily today

1990-1999

- Visual Basic (1991)
- Ruby (1993)
- Java and JavaScript (1995)
- PHP (1995)

2000-2010

- C# (2001)
- Windows PowerShell (2006)

Programming Levels

Low-Level Programming

- Runs as close as possible to the hardware
- May include operating systems, drivers, or direct access to hardware, such as video card parallel processors

High-Level Programming

- Runs 'on top' of a low-level program.
- Don't need to worry about coding many drivers, graphics, etc.

Types of Programming Languages



Scripting Languages

Scripting Languages

- bash
- PowerShell

Data-Oriented Languages

- SQL, MySQL
- dBase

Types of Languages

Embedable Languages

- PHP
- Perl
- Ruby
- JavaScript
- Java
- VBScript

Compiled Languages

- C
- C++
- C#
- COBOL
- Java
- Objective-C
- Python
- Rust
- Swift
- Visual languages

Interpreted Languages

- J
- JavaScript
- Perl
- PHP
- Python
- Ruby
- VBScript
- Windows PowerShell

Other Languages

The following are *not* programming languages

- HTML: Web-based markup language
- CSS: Design-based style sheet language
- XML: Document encoding markup language

ArnoldC, a language devised by Finnish computer programmer Lauri Hartikka, assigns programming functions to catchphrases from Arnold Schwarzenegger movies. Some keywords:

- If: BECAUSE I'M GOING TO SAY PLEASE
- While: STICK AROUND
- MultiplicationOperator: YOU'RE FIRED
- DivisionOperator: HE HAD TO SPLIT
- DeclareMethod: LISTEN TO ME VERY CAREFULLY
- EndMethodDeclaration: HASTA LA VISTA, BABY
- AssignVariable: GET TO THE CHOPPER
- Return: I'LL BE BACK

“At the end whether you are scripting, coding or programming, you are providing some environment like a PC, robot, or browser, a set of instructions on what tasks it needs to perform. Whether the language you do this is high level or low level has nothing to do with it. Whether it ends in “script” or ends in “++” has nothing to do with it. Whether it runs on your browser or on a mainframe has nothing to do with it. Whether it is intended to be sold to millions or just used by you has nothing to do with it.

I’d suggest you use none of the above terms. You’re a software craftsman. You design and build solutions using the best suited technologies for the problem you are trying to solve.”