

# CS100: CPADS

# Programming Fundamentals

---

David Babcock / James Moscola  
Department of Physical Sciences  
York College of Pennsylvania



# Programming 101

---

- **Programming is about problem solving and algorithm design**
- **Programming is NOT simply writing code**
- **Writing code is only a small part of programming**
  
- **A program is a set of sequential instructions to complete a task along with the necessary data structures to represent the information**

# Programming 101

---

- **There are a number of steps in the development of a program:**
  - (1) **Requirement specification** - What do you want the program to do?
  - (2) **Analysis** - How can those requirements be broken down into specific tasks?
  - (3) **Design** - Determine the best way to construct a program to complete those tasks. Determine how to integrate those tasks satisfy the original requirements.
  - (4) **Coding** - Implement your design using an appropriate programming language (not all programming languages are well-suited for all problems)
  - (5) **Testing/Debugging** - Validate that your program operates correctly and produces the correct output.
  - (6) **Deployment** - Provide a completed product to the end user.

# Programming 101

---

- **Programming languages have formal syntax and are designed to be unambiguous**
  - Programs do exactly what you tell them to do, though not necessarily what you want them to do
- **Unlike natural languages that can be very ambiguous**
  - KIDS MAKE NUTRITIOUS SNACKS
  - STOLEN PAINTING FOUND BY TREE
  - HOSPITALS ARE SUED BY 7 FOOT DOCTORS
  - ENRAGED COW INJURES FARMER WITH AX

# Types of Programming Languages

---

- **Compiled Languages**

- Text-based source code is converted into machine executable code prior to deployment
- Compiled languages are architecture specific
- Can be optimized for performance by the compiler
- Examples include: C, C++

# Types of Programming Languages

---

- **Interpreted Languages**

- Text-based source code is converted into machine executable code when program is run
- Interpreted languages are independent of architecture
- Not typically used for performance oriented programming (conversion from text-based code to machine code takes time)
- Examples include: Python, AWK, Perl, JavaScript

# Types of Programming Languages

---

- **Mixed Languages**

- Text-based source code is converted in an intermediary type of code called bytecode prior to deployment
- Bytecode is typically architecture independent and is intended to be portable
- Bytecode can be optimized
- Bytecode is interpreted by a Virtual Machine when the program is run
- Examples include: Java