Computational Expression

Computer and Java Basics

Janyl Jumadinova

4 September, 2019
What is Computer Science?

- A computation is a sequence of well-defined operations that lead from an initial starting point to a desired final outcome.
A computation is a sequence of well-defined operations that lead from an initial starting point to a desired final outcome.

Computer science is the study of computation.

Computers are a medium for expression in computer science.
What is a computer?
Digital Computers

- Information is broken into discrete pieces.
- Each piece is represented as a number, a **binary number**.
Information is broken into discrete pieces.

Each piece is represented as a number, a binary number.

Binary Numbers:

- Base 2 system.
- A single bit: either 0 or 1
  - can represent two items, e.g. state of the light bulb.
- $N$ bits can represent $2^N$ unique items.
What is a computer?

- Slow deployment times
- Huge costs
- Wasted resources
- Difficult to scale
- Difficult to migrate
What is a computer?

- Slow deployment times
- Huge costs
- Wasted resources
- Difficult to scale
- Difficult to migrate
IT Landscape is Changing

- Apps
- Cloud
- DevOps
IT Landscape is Changing

- Migrate workloads to cloud
- Portability across environments
- Want to avoid cloud vendor lock-in
Container-based Approach
Container-based Approach

Containers are an app level construct
Docker

- **14M** Docker Hosts
- **900K** Docker apps
- **77K%** Growth in Docker job listings
- **12B** Image pulls
- **3300** Project Contributors

Janyl Jumadinova

Computational Expression

4 September, 2019
Docker
Now the Bad News

In this class you will not learn how to make Docker images, containers, etc. - you are only using Docker to run your programs. More bad news!

Docker does not support Windows Home very well.

Plan for Windows Home:
install software (below) directly on the machine - not use Docker.

- Will do this in the lab tomorrow, in addition to learning how to use Git.

Mac, Windows Pro/Enterprise will still use Docker Desktop - no need to install anything else.

Now we begin our adventure into programming!
Now the Bad News

- In this class you will not learn how to make Docker images, containers, etc. - you are only using Docker to run your programs.

Plan for Windows Home:
- install software (below) directly on the machine - not use Docker.
- Will do this in the lab tomorrow, in addition to learning how to use Git.

Mac, Windows Pro/Enterprise will still use Docker Desktop - no need to install anything else.
Now the Bad News

- In this class you will not learn how to make Docker images, containers, etc. - you are only using Docker to run your programs.
- **More bad news!** Docker does not support Windows Home very well.
Now the Bad News

- In this class you will not learn how to make Docker images, containers, etc. - you are only using Docker to run your programs.
- **More bad news!** Docker does not support Windows Home very well.

**Plan for Windows Home:** install software (below) directly on the machine - not use Docker.
- Will do this in the lab tomorrow, in addition to learning how to use Git.
  - Mac, Windows Pro/Enterprise will still use Docker Desktop - no need to install anything else.
Now the Bad News

- In this class you will not learn how to make Docker images, containers, etc. - you are only using Docker to run your programs.
- **More bad news!** Docker does not support Windows Home very well.

**Plan for Windows Home:** install software (below) directly on the machine - not use Docker.
- Will do this in the lab tomorrow, in addition to learning how to use Git.
  - Mac, Windows Pro/Enterprise will still use Docker Desktop - no need to install anything else.
- Now we begin our adventure into programming!
What is Computer Programming?

Programming is the act of writing usable and useful software. A program is a set of instructions.
What is Computer Programming?

- Programming is the act of writing usable and useful software
- A program is a set of instructions
We will use **Java** programming language in this class.

Java is a programming language originally developed by Sun Microsystems and released in 1995 as a core component of Sun’s Java platform.
HISTORY OF JAVA

• Started development in 1991 at Sun
• Originally called Oak
• Intended for smart consumer-electronic devices
• Derives much syntax/concepts from C++
• BCPL → B → C → C++ → Java
• Development almost halted, but 1993 saw introduction of web; Java was revamped to be able to easily add dynamic content to web pages
• Formally announced and released in May 1995
• Released under GPL to the public in May 2007
Programming in Java

- Java is an **object-oriented** programming language
Java is an object-oriented programming language

Objects are fundamental elements that make up a program
Programming in Java

- Java is an object-oriented programming language
- Objects are fundamental elements that make up a program
- Java has a library of software, called Java API, that is available for your use
Java program development process
Simple first Java Program: “Hello World”

/** This is the first program people write in a new language, the "Hello World!". In Java, this file must be named Welcome.java, with the first part of the name, Welcome, being the same as the name of the class. The filename itself (not the class name) must always end in .java to indicate to the operating system that it’s a java source file.
*/

public class Welcome {
    public static void main ( String args[] ) {
        System.out.println ( "Hello World!" );
    }
}

For today, try running this on https://www.jdoodle.com/online-java-compiler/
Comments in Java can be one of three styles:

- **Single line:** starts at `//` anywhere on a line, ends at the end of that line
- **Multi-line:** starts with character sequence `/*` anywhere, ends with character sequence `*/` anywhere after that can span multiple lines
- **javadoc:** starts with character sequence `/**` anywhere, ends with character sequence `*/` anywhere, after that uses javadoc utility to create HTML documentation from code
**public class Welcome:**

- **public** means that something is available across packages (reserved word)
- Name of the class has to be the same as the name of the .java file
public class Welcome:
  
  **public** means that something is available across packages (reserved word)
  
  Name of the class has to be the same as the name of the .java file

public static void main ( String identifier[]):

  The particular form of main is required by Java.
  JVM starts executing here!
  main is a static method, it is part of its class and not part of objects.
  Strings in Java are sequence of characters
public class Welcome:
  
  public means that something is available across packages (reserved word)
  
  Name of the class has to be the same as the name of the .java file
  
  public static void main ( String identifier[]):

  The particular form of main is required by Java.
  JVM starts executing here!
  main is a static method, it is part of its class and not part of objects.
  Strings in Java are sequence of characters

  Braces { } are used to collect statements into a "block"
public class Welcome:
  
  public means that something is available across packages (reserved word)
  Name of the class has to be the same as the name of the .java file

public static void main ( String identifier[]):
  The particular form of main is required by Java.
  JVM starts executing here!
  main is a static method, it is part of its class and not part of objects.
  Strings in Java are sequence of characters

  Braces { } are used to collect statements into a ”block”

  Statements in Java end with semicolons.
Printing

- **println**: New line after printing
- **print**: No new line
- **printf**: Can specify format - may learn this later
string literal in class String
“ABC”
“This is interesting”
“ ”
“91”
**Character Strings**

string literal in class **String**

“ABC”

“This is interesting”

“   ”

“91”

- Use **print** or **println** methods to print a character string to the terminal
- **System.out.println(“CMPSC 111”);**
- the string **“CMPSC 111”** is a **parameter**: data sent to a method