

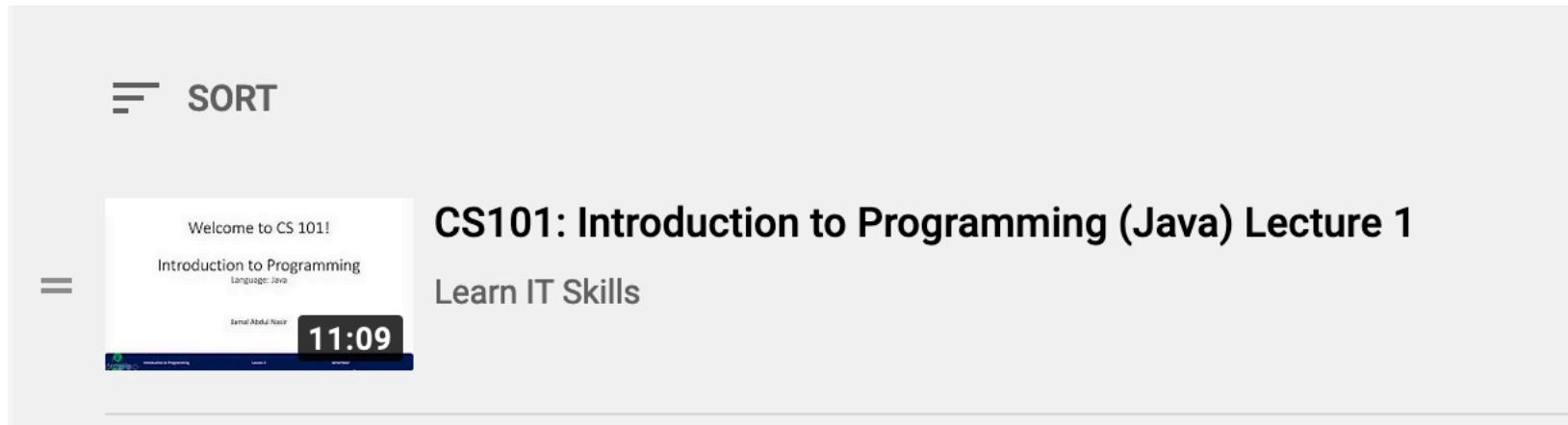
Welcome to CS 101!

Introduction to Programming (in Java)

Jamal Abdul Nasir

Previous Lectures to watch?

Playlist name: Introduction to Java Programming



Course Learning Objectives

- Computational Thinking (Decomposition into sub-problems)
- Functionality/Behavior (Working solution of these sub-problems)
- Control Structures (Methods, loops, conditions)
- Data Abstraction (Variables, parameters, arrays, classes)
- Code Quality (programs that are well-written, readable, maintainable, and conform to established standards)

Complexity

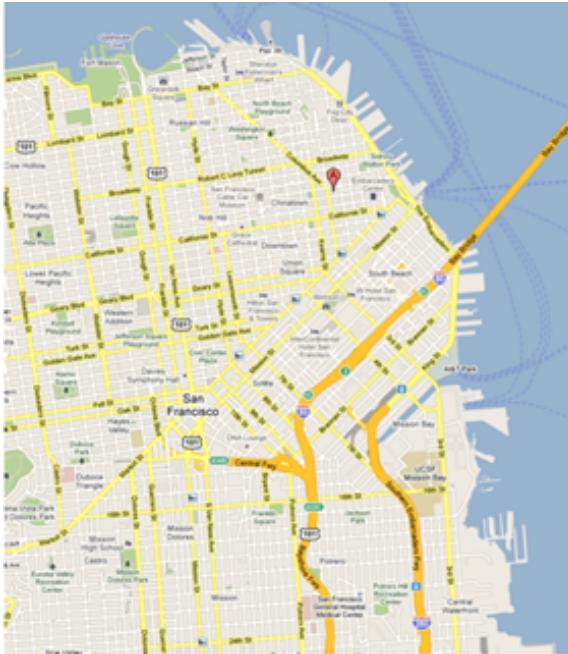
- Programming is straightforward, as long as your programs are small
 - *Complexity* is our enemy
 - *Abstraction* is the key to conquering complexity

Abstraction

- **Abstraction** allows us to build general-purpose artifacts
 - **Detail Removal:** Hide unnecessary details from users and designers
 - **Generalization:** Avoid unnecessary repetitive work
- Learning to reason using the most appropriate abstraction is a key goal of computational thinking

Abstraction (Detail Removal)

- “The act or process of leaving out of consideration one or more properties of a complex object so as to attend to others.”



Maps for directions

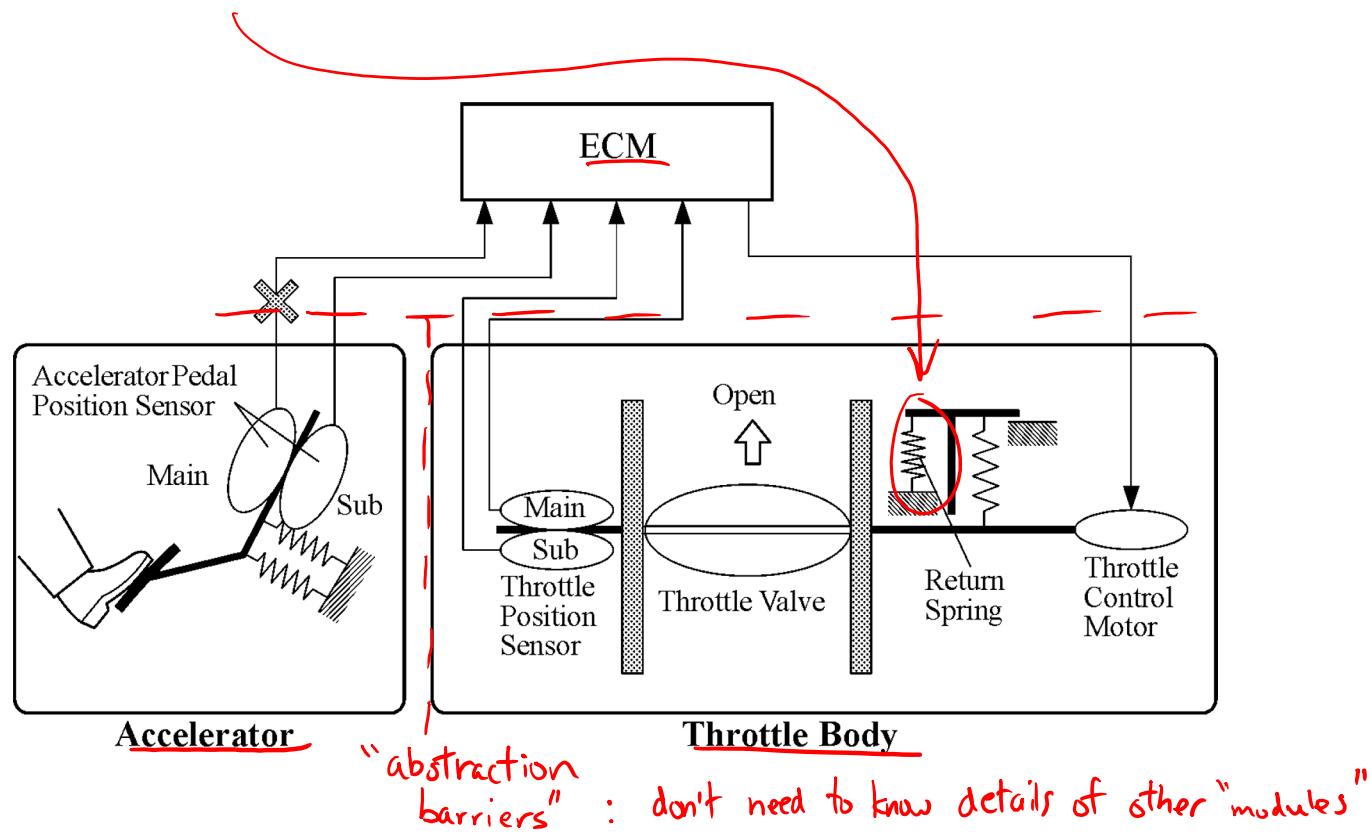
Abstraction (Detail Removal)

- Modern user interface: Right pedal is “accelerate”, left is “decelerate”



Maps for directions

Abstraction (Detail Removal)



Abstraction: Generalization

- “The process of formulating general concepts by abstracting common properties of instances.”
- Extensible shower rods
- Adjustable hats
- Single recipe for <fruit> milkshake
- Feeding animals on a farm
 - To feed <animal>, put <animal> food in <animal> dish

Abstraction: Generalization

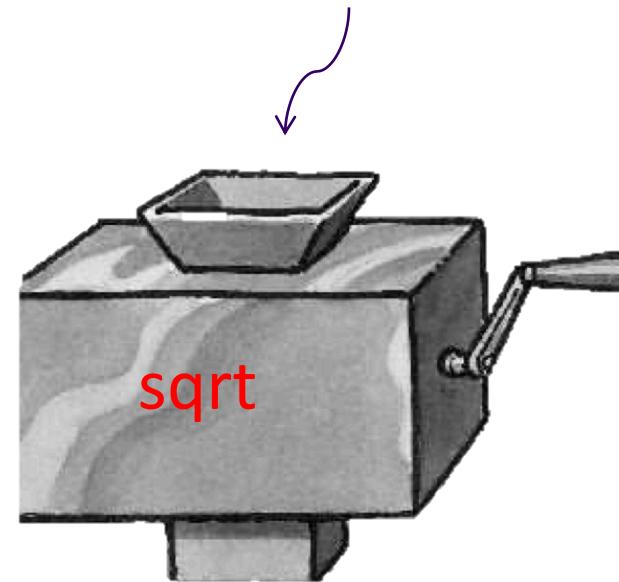
- You are going to learn to write functions, like in math class:

$$y = \text{sqrt}(x)$$

- x is the argument

Abstraction: Generalization

25.0



5.0

Abstraction (Summary)

- Abstraction is one of the most important challenges in computer science
 - How do you identify the right abstraction you need (block to build) to solve your problem?
- Think about computers:
 - How many of you actually know how a computer works?
 - How many of you can use a computer?
 - Thanks to abstraction!!!

Algorithm

- Computational thinking is to divide problems into sub-problems
- Then algorithm is a **step-by-step** procedure for solving each sub-problem

What is programming?

- **program**: A set of instructions to be carried out by a computer.
- **program execution**: The act of carrying out the instructions contained in a program.
- **programming language**: A systematic set of rules used to describe computations in a format that is editable by humans.
 - We will be studying a programming language called Java



Programming is like Legos...

With programming we can make anything



Building Java Programs

Your first Java program!

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

- File must be named `Hello.java`

Running a program

- Write it
- Compile it
- Execute it

Running a program

Write it:

- **code or source code:** The set of instructions in a program
- *Compile* it.
 - **compile:** Translate a program from one language to another.
 - **byte code:** The Java compiler converts your code into a format named *byte code* that runs on many computer types.
- *Run (execute)* it.
 - **output:** The messages printed to the user by a program.

Running a program

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



Bigger Program

```
public class Hello {  
    public static void main(String[] args) {  
        System.out.println("Hello, world!");  
        System.out.println();  
        System.out.println("This program produces");  
        System.out.println("four lines of output");  
    }  
}
```

Hello, world!

This program produces
four lines of output

Structure of a Java program

```
public class name {  
    public static void main(String[] args) {  
        statement;  
        statement;  
        ...  
        statement;  
    }  
}
```

class: a program

method: a named group of statements

statement: a command to be executed

Structure of a Java Program

- Every executable Java program consists of a **class**,
 - that contains a **method** named `main`,
 - that contains the **statements** (commands) to be executed.

Names

- You must give your program a name.

```
public class HelloWorld {
```

- Naming convention: capitalize each word (e.g. `HelloWorld`)
- Your program's file must match exactly (`HelloWorld.java`)
 - includes capitalization (Java is "case-sensitive")

Identifier

- A name given to an item in your program.
- It must start with a letter or _ or \$
- Subsequent characters can be any of those or a number

Legal Identifier

- _myName
- TheCure
- ANSWER_IS_42
- \$bling\$
- Age
- Full_name

Illegal Identifier

- me+u
- 49ers
- side-swipe
- Ph.D's

Keywords

- **keyword:** An identifier that you cannot use because it already has a reserved meaning in Java.

abstract	default	if	private	this
boolean	do	implements	protected	throw
break	double	import	public	throws
byte	else	instanceof	return	transient
case	extends	int	short	try
catch	final	interface	static	void
char	finally	long	strictfp	volatile
class	float	native	super	while
const	for	new	switch	
continue	goto	package	synchronized	

Keywords in a Java program

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

Keywords in a Java program

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

Keywords

- Because Java is case-sensitive, you could technically use Class or cLass as identifiers, but this is very confusing and thus **strongly discouraged**.

Revision

- Complexity vs. Abstraction
- Abstraction (Detail Removal + Generalization)
- HelloWorld program in a Java
- Structure of a Java Program
- Names and Identifiers
- Keywords

The End

- Practice makes a man perfect!
- If you have questions/comments: Comment on Youtube Video.
- Please share with your friends and keep Learning!!! Bye!