

**02-0: Java Programs**

- Java programs are a collection of classes
  - Each class is a *Template*, not an *Object*
  - Can't use a class until we create an instance (call "new")
- One exception: Static Methods
- In this course, we will only use one static method: main method

**02-1: Java Programs**

- Simplest Java Program:
  - Hello World 3 ways

**02-2: Classes**

- Classes contain data and methods (like functions in python)
- Methods can access data members (called instance variables) of a class
- Special method, called "Constructor", which is called when an object of the class is created using new.

**02-3: Classes**

- Second Example: Employee

**02-4: Implementation I: Locals**

- Variables declared inside functions are called local variables
- Stored on the *Call Stack*
- Created when function is called
- Disappear when function ends

**02-5: Implementation I: locals**

- When a method is called:
  - Allocate space on the call stack to store method parameters & variables local to the method
    - Activation record for the method
  - Copy values into the space allocated for the parameters
  - Execute the body of the method
  - Pop activation record off the stack

**02-6: Implementation I: locals**

- Example: MethodTest
  - Code
  - Memory contents (on whiteboard)

**02-7: Pass-By-Value**

- Methods in Java are “Pass-By-Value”
- Value is copied into the parameter
- Changes to the parameter don’t change the value passed in
- Example: MethodTest2

**02-8: Objects**

- Stack is not the only place where data is stored
- Objects (specifically, instance variables) are stored on the Heap
  - Different section of memory than the stack
  - Memory is *allocated* (or set aside) on the heap through a call to **new**

**02-9: Objects**

- HeapData, HeapDriver1

**02-10: Objects**

- Every object in your program is a pointer to memory on the heap
- When we pass an object as a parameter, copy the pointer
- Changes to the *pointer itself* are not reflected back to the calling method
- Changes made to *what the pointer points to* are reflected back to the calling method

**02-11: Objects**

- HeapData, HeapDriver2, MethodTest3