204700
Data Structure and Programming Languages

Jakarin Chawachat

From: http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/index.htm

Outline

- Types, Variables, Operators
- More Types, Methods, Conditions
- Loops, Arrays
- Classes and Objects
- Data structures

Course Description

- Abstract data types
- Linear data structure
- Non-linear data structure
- Searching and sorting techniques
- Programming language paradigms

Installing Java and Eclipse

In order to write Java programs, you need

- The Java Development Kit(JDK)
  - Contains the tools needed to compile and run Java programs
- A source code editor
  - Lets you write programs and has feature to make this easier

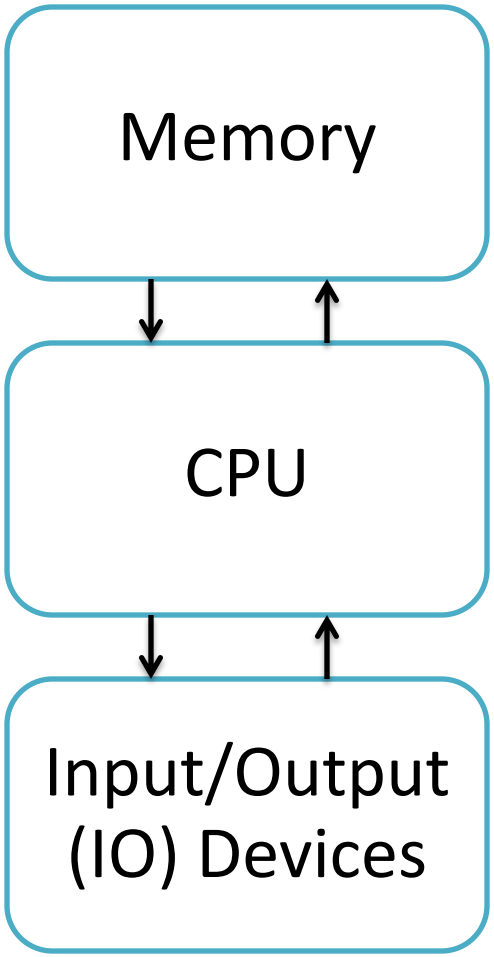
Installing Java and Eclipse: Editors

In order to write programs needs a piece of software called an editor

- Source code editor
  - Notepad, SciTE, UltraEdit, Textmate
- Integrated Development Environments(IDEs)
  - Eclipse, Netbeans

1.TYPES, VARIABLES, OPERATORS

The Computer



CPU Instructions

z = x + y	**Read location x** **Read location y** **Add** **Write to location z**

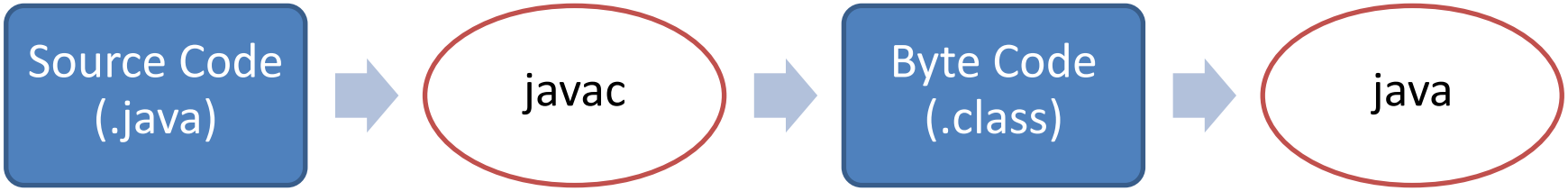
Programming Languages

- Easier to understand than CPU instructions
- Needs to be translated for the CPU to understand it

Java

- “Most popular” language
- Runs on a “virtual machine” (JVM)
- More complex than some (eg. Python)
- Simpler than others (eg. C++)

Compiling Java



Compile: javac filename.java → filename.class
Run: java filename

Example:

javac hello.java → hello.class
java hello

First Program

```
class Hello{
    public static void main(String[] arguments){
        //Program execution begins here
        System.out.println("Hello world.");
    }
}
```

Program Structure

```
class CLASSNAME{
    public static void main(String[] arguments){
        STATEMENTS
        ……
    }
}
```

Output

System.out.println(some String) outputs to the console

Example:

```
System.out.println("output");
```

Second Program

```
class Hello2{
   public static void main(String[] arguments){
        System.out.println("Hello world.");
        System.out.println("Line number 2");
   }
}
```

Types

Kind of values that can be stored and manipulated.

- **boolean**: Truth value (**true** or **false**).
- **int**: Integer (0, 1 -50)
- **double**: Real number (3.14, 1.0, -756.015)
- **String**: Text (“Hello world.”, “example”).

Variables

Named location that stores a value of one particular type.

Form:

TYPE NAME;

Example:

String foo;

Assignment

Use = to give variables a value.

Example:

```
String foo;
foo = “Java programming”;
```

Assignment

Can be combined with a variable declaration.

Example:

```
double badPi = 3.14;
boolean isJanuary = true;
```

```
class Hello3{
    public static void main(String[] arguments){
        String foo = “Java Programming”;
        System.out.println(foo);
        foo = “204700”;
        System.out.println(foo);
    }
}
```

Operators

Symbols that perform simple computations

Assignment: =

Addition: +

Subtraction: -

Multiplication: *

Division: /

Order of Operations

Follows standard math rules:

1. Parentheses ()
2. Multiplication and division * /
3. Addition and subtraction + -

```
Class DoMath{
    public static void main(String[] arguments){
        double score = 1.0 + 2.0 * 3.0;
        System.out.println(score);
        score = score / 2.0;
        System.out.println(score);
    }
}
```

```
Class DoMath2{
    public static void main(String[] arguments){
        double score = 1.0 + 2.0 * 3.0;
        System.out.println(score);
        double copy = score;
        copy = copy / 2.0;
        System.out.println(copy);
        System.out.println(score);
    }
}
```

String Concatenation (+)

```
String text = “hello” + ”world”;
Text = text + “ number ”+5;

//text = “hello world number 5”
```

Assignment: C2FCalculator

Convert Celsius degree to Fahrenheit degree:

$$\frac{c}{5} = \frac{f - 32}{9}$$