APCS Notes

Yicheng Wang

2014-2015

Contents

1 2014-9-8

1.1 Comparison of Programming Languages

Scheme:
Annoying Prefix Notation
strict syntax
not object oriented
only seperated by parenthesis
IDE NOT necessary
Mostly recursion + list
Functional Programming Language (Everything is a function)

Netlogo:
GUI based
Shines on Interactive Modeling
Bad for input/output data
Parallel Programming Language
Not a general-purpose language – ONLY USEFUL IN NETLOGO ENVIROMENT
Netlogo IDE (Integrated Development Enviroment) NECESSARY

Python:
High-level Language
Uses Indentation
infixed math + prefix function
Linear processing
General-purpose language
Interperted Language
IDE NOT necessary

Java:
Object oriented
infixed math + prefix function
Mid-level Language

LaTeX
Markup Language
Compiled Language

2 2014-9-10

2.1 "Hello World" in Java

Last year, we learned how to write the "Hello World" program in python.

```
def hello():
    print "Hello world!"

hello()
```

Now, we'll learn about how to do this in Java:

Java is more restrictive than python, java programs are usually in their own folders. Java's invented for portability and "amount of stupid/super-smart people." Different smart people have different ways of approaching problems. Java's designed to limit people's ways of doing things to make big project easy. Real good programmers don't like java... b/c it's restrictive. Java is designed to be industrially viable.

An object defines a specific thing within your program. Everything in java is an object.
A Class = object type.
Tradition = 1 class per file, named starting with upper-case letter

Here's a simple program in Java:

```
/*
    This is a null line
    C'est un comment!
*/

// C'est un end-of-line comment

import java.io.*;
import java.util.*;

public class Hello {  // public = the outside world (aka other things in your program) can see this
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

2.2 Running Java

Source code (foo.java) → Java compiler (foo.class) → JVM
Note that java doesn't compile to machine code, java compiles to javaBiteCode using JVM. This is where the portability comes in.

2.3 Java technicalities

method = function in python
You need a method in one of you're classes called "main"

3 2014-9-11

3.1 Moving into the "java way" of doing things!

Java is object oriented. Object oriented means that the world is made of objects. Every object has its unique attributes. Objects also have abilities, aka things they can do. Every program in java is made of objects.

Let's take the example of a simple chess program. An example of an object would be a pawn. It would have attributes like color and position. Its abilities would include moving and attacking. However, these pawns are different, White pawn 1-8 and Black pawn 1-8. They behave in the same way, but they have different positions. You don't want 16 seperate definitions b/c most of them are the same. Therefore one would create a class for all of the pawns, which would define the "info about objects." We then make objects which are known as "instances of a class." Objects are made based on the definitions defined within the class.

Hello world program 2 – the java way:

```
1  // We'll use objects to do stuff
2
3  import java.io.*;
4  import java.util.*;
5
6  public class Greeter {
7      // We put the attributes here
8
9      // We put the abilities here
10     // In Java, these are called methods
11     // Methods are functions, but they belong to specific classes
12     public void greet() {
13
14     // public = can be called from outside the class
15     // void = this doesn't send anything back, like null returner in C
16
17         System.out.println("Hello world!");
18     }
19 }
```

4 2014-9-15

4.1 Typical anatomy of Java Program

A program is consisted of objects. One must tell java where to start the program –¿ public static void main One calls that class "driver," it starts the java program.

Driver.java:

```
1
2  import java.io.*;
3  import java.util.*;
4
5  public class Driver {
6      public static void main(String[] args) {
```

```
7
8         //How to use the greeter within the driver.
9
10        Greeter g;
11        //Creates a local variable to be of type greeter
12
13        /*
14        Variable declaration, all variables must be declared
15        like global, turtles-own and patches-own variables in netlogo
16        Declaration specifies the type of the variable
17        local variable = a variable only visible/usable within a method,
      created when the method is called, destroyed when the function exits
18        */
19
20        /*
21        When main is ran, it occupies some memory on the computer
22        Greeter g is a small box within main, we need to do something with it
23        or java refuses to do stuff with it
24        */
25
26        g = new Greeter();
27        /*
28        New:
29         1. Allocates enough memory to store a Greeter.
30         2. Do whatever's necessary to setup / initiates the memory to be a
      Greeter.
31         3. Returns the address of the memory that was allocated.
32
33         The assignment statement stores the address in g.
34        */
35
36        System.out.println(g);
37
38        // This prints the location of the variable g within the memory
39
40        /*
41        When this file is compiled, Greeter is compiled as well
42        All methods/class called during main are compiled as well
43        */
44
45        g.greet();
46        /*
47        Accesses the greet method within the class g.
48        */
49    }
50 }
```