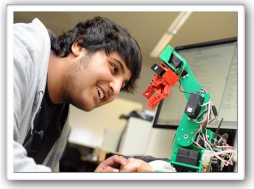


Programming Concepts

Department of Computer Science
University of Warwick

Language Choices

There are thousands of programming languages - just notation

We group them by their commonalities and properties

- High-level vs. low-level
- General purpose vs. special purpose
- Compiled vs. interpreted

Implications for students learning computer programming

Perspectives

Technical

- All languages are in some sense equal - we can show it!

Practical

- A first language is important, it's a gateway to understanding

Personal

- I'm confident in teaching a language so it's a good choice

Computing Is Not Just Programming

Computer programming can be a gateway to understanding

Studying programming is not the same as studying computing

What Does All That Mean For Programming?

Algorithmic thinking is more valuable than knowing any language

Languages of particular types share common characteristics

A good set of examples - Python, Java, C, C++...

See For Yourself - Java vs. Python

Input and Output

Variables and Data Types

Constructs - Control Flow, Conditionals and Iteration

Arrays and Data Structures

Functions / Methods

Input and Output

Python

```
import sys

for arg in sys.argv:
    print arg
```

Java

```
public static void main (String[] args) {
    for (String s: args) {
        System.out.println(s);
    }
}
```


Variables and Data Types

Python

```
width = 100

myPi = 3.14

check = True
```

Java

```
int width = 100;

double myPi = 3.14;

boolean check = true;
```

Constructs - Control Flow

The ordering of statements is similar to many high-level languages

Sequential execution of statements governed by structures

Python

```
width = 100

length = width * 3

area = length * width
```

Java

```
int width = 100

int length = width * 3

int area = length * width
```

WARWICK

Constructs - Conditionals

Python

```
if x == 1:
        print ‘One’
elif x == 2:
        print ‘Two’
else:
        print ‘Unknown’
```

Java

```
if (x == 1) {
        System.out.println(“One”);
} else if (x == 2) {
        System.out.println(“Two);
} else {
        System.out.println(“Unknown”);
}
```

WARWICK

Constructs - Iteration

Python

```
for i in range(0,3):
        print i

while (count < 5):
    print count
    count = count + 1
```

Java

```
for (int i = 0; i < 3; i++) {
        System.out.println(i);
}

while (count < 5) {
    System.out.println(count); count++;
}
```

WARWICK

Arrays and Data Structures

Python

```
list = [‘one’,’two’,’three’]

list[1]
```

Java

```
String[] list = new String[3];

list[0] = “one”;

String[] list = {“one”,”two”,”three”};
```

WARWICK

Functions / Methods

Python

```
def add(one,two):
       return one + two

def print(str):
        print str
```

Java

```
public int add(int one, int two) {
       return one + two;

public void print(String statement) {
       System.out.println(statement);
}
```


What Is Important

Problem solving

How can we define a process to find prime factors, e.g, factors of 140?

140	Is 140 evenly divisible by 2? Yes! Remember 2 and divide 140 by 2.
2 * 70	Is 70 evenly divisible by 2? Yes! Remember 2 and divide 70 by 2.
2 * 2 * 35	Is 35 evenly divisible by 2? No, how about 3? No. 4? Nope. 5? Yes! Remember 5 and divide 35 by 5.
2 * 2 * 5 * 7	And we're done!

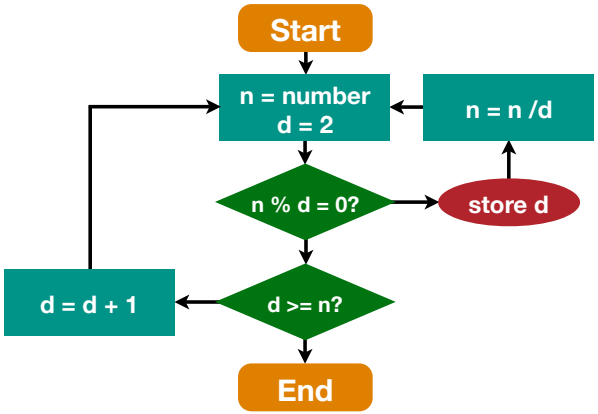
What Isn't Important

```
let primeFactors n =
  let inline isFactor n d = n % d = 0L

  let rec nextFactor n d =
    let x = if d = 2L then 3L else d+2L
    if isFactor n x then x else nextFactor n x

  let rec findFactors n d acc =
    if isFactor n d then
      findFactors (n/d) d (d::acc)
    elif n > d then
      findFactors n (nextFactor n d) acc
    else
      acc
```


What Would Be Ideal



Agree or Disagree?

Programming languages are a necessary notation

Problem solving and rigorous thinking are the heart of Computing

Programming is a gateway that pupils can (should?) be helped through