

Introduction to Programming

A quick overview of Java, Object-Oriented programming and the programming process

1

Overview

- Java: the programming language
- Object Oriented programming
- Objects: attributes and methods
- Creating Objects
- Programs
 - source file, compilers, bytecode, execution

2

Java: the programming language

- A computer programming language
- About 20 keywords
- Strict syntax rules
- Invented last decade (about 1993)
- Object-oriented programming language

3

Object Oriented Programming Basics

- We start with the fundamentals of Object Oriented Programming
- Then, we apply these principles in a robot programming environment
- Then, we move on to standard programming in Java

4

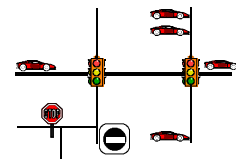
The World of Objects

- The World consists of Objects
- Objects are Nouns
- When we write a program objects are generally used to represent real world objects.

5

Example: Traffic System

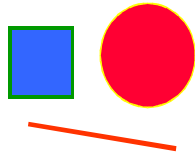
- Simulate traffic flow, traffic jams
- Objects include:
 - Cars
 - Trucks
 - Pedestrians
 - Traffic Lights
 - The Road itself!



6

Graphical Drawing System

- Allow user to draw shapes and manipulate them on the screen
- Objects include:
 - Circles
 - Rectangles
 - Lines



7

Objects have a State — Attributes

An *attribute* is any characteristic of an object, e.g. colour, shape etc.

8

Objects Can *Do* Things — Methods

An object has operations it can perform — built right into it, e.g. move

9

Basic Objects

- **Objects**
 - Nouns, things in the world
- **Attributes**
 - Properties each of these things have
- **Methods**
 - Actions that each of these things can do

10

Let's Consider Shapes

- Shapes have a state — attributes
- Attributes of a shape:
 - Filled, line width, line color, fill color, location
- Shapes can *do* things — methods
- Methods of a shape:
 - Fill, Empty, Move, Grow

11

Each Shape is an Object

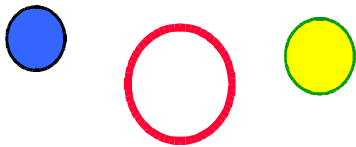
A green circle, a red square, a yellow arrow, and a pink star. A blue line points from the text to the shapes.

- Properties of a shape:
 - filled
 - line width
 - line color
 - fill color
 - location
- Methods of a shape:
 - Fill
 - Empty
 - Move
 - Grow

12

There is a Structure Here

- There are certain shapes of a related kind
- This *prototype* is called a Class



- Each circle is different, but they are all instances of the class Circle

13

Each Object is an *Instance* of a *Class*

- An Instance of the Class “Circle”



- Two Instances of the Class “Square”



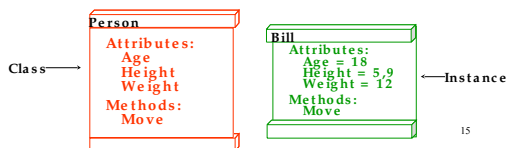
- An Instance of the Class “Line”



14

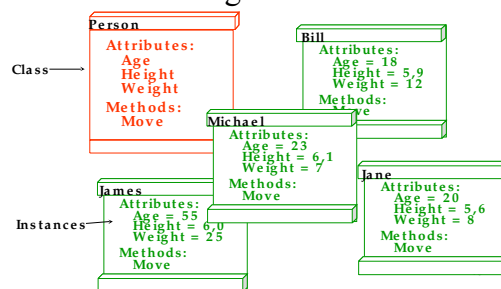
Classes

- A *Class* is an abstract description of objects having the same attributes and methods
- A specific Object is an instance of a Class
- A Class is the cookie cutter — An Object is the cookie



15

Many Different Objects from a Single Class

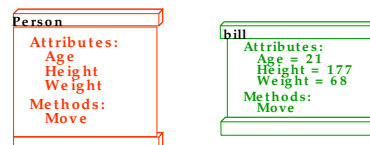


How Do We Create an Object?

- We use a *constructor*
- This takes a Class and creates an Instance of the class, an object, perhaps with certain properties
- “Construct an Instance of the Class Person, give it the name “bill”, and make its Age be 18, its height be 5 foot 9, and its weight be 12 stone.”

17

How Do We Create an Object?



- Presto! We now have an object “bill”, with certain attributes, and with the method Move
- Bill’s Move method can now be executed.

18

Object Vocabulary

- **Class**: Definition of an object
- **Objects**: Nouns, things in the world.
An Object is an instance of a Class.
- **Constructor**: Creates an Object from a Class.
- **Attributes**: Properties of an object.
- **Methods**: Actions that an object can do

19

Objects and Java

- Java has a keyword **class** which is used to create an object prototype.
- Every Java program defines at least one class
- Here is an example

```
class House
{
}
```

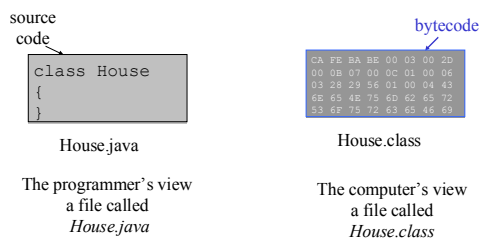
House.java

This class has no attributes or methods and is not useful. It is only used as an example

20

So what exactly is a program?

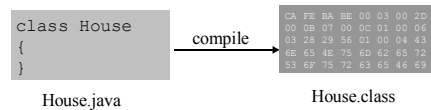
There are two views of a program



21

Compiling a Java Program

- The class file is generated from the source code by a compiler.



- You can run the compiler as shown below

The compiler is called javac

```
H:\> javac House.java
```

22

Steps to a program

- Create the program file, e.g. *House.java*.
- Compile the file *House.java* to create the file *House.class*.

```
H:\> javac House.java
```

- Check that the file *House.class* was created (use the directory command)

The DIR command shows the newly created class file

```
H:\> dir
13:54         17 House.java
13:55        184 House.class
H:\>
```

23

Objects from classes

- Use **new** to create an instance from a class definition.
- The instruction **new House()** creates a new instance of the House class.
- An instance isn't much use without a name; a name is assigned to an instance using the following instruction
House whiteHouse = new House();
This creates a new instance of the House object and calls it **whiteHouse**.

24

A program that does something!

- Here is a class which prints something

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

Hello.java

- When this is compiled it can be run by the **java** program

```
H:\>javac Hello.java
H:\>java Hello
Good Morning
H:\>
```

The program
prints Good
Morning

25

And that's Programming

1. Create the source code
2. Create the class file (by compiling the source)
3. Run the program. You use the **java** program to execute the class file.

Try this on your computer.

26