

The background of the slide is a reproduction of a painting, likely Vincent van Gogh's 'Wheat Field with Oxcarts'. It depicts a vast, golden-yellow wheat field under a blue sky. In the foreground, there are several large, rounded haystacks. A lone figure, a person in a white shirt and dark trousers, stands in the middle ground, looking towards the right. In the background, there are several simple buildings with red roofs and a few trees. The overall style is characteristic of Impressionism, with visible brushstrokes and a vibrant color palette.

J01 Introductory Java 1

Imperative programming languages

Java standard library

Types

Hello world

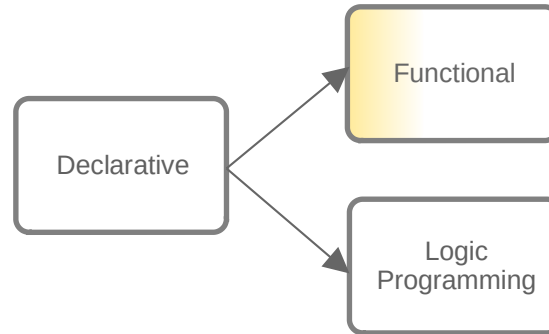


Why Java?

- Learn multiple programming paradigms
- Important example of:
 - Object-oriented programming
 - Large scale programming
 - Programming with a rich standard library

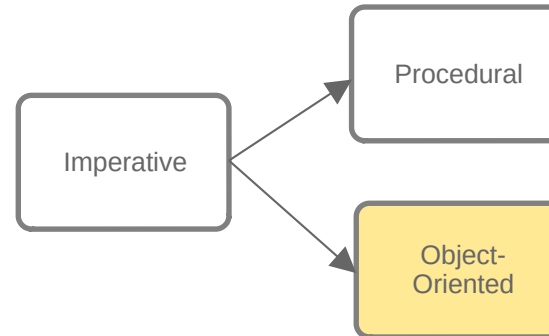
Programming Paradigms

Declarative programming describes the desired result without explicitly listing steps required to achieve that goal.



Pure functional programming only transforms state using functions without side effects.

Imperative programming describes computation in terms of a series of commands that transform state.



Procedural programming uses procedures to transform data structures

Object-oriented programming tightly groups the procedures and data structures together into “objects”

Structured Programming

Another paradigm that imposes a **logical structure to code** making it easier to understand and less error prone.

- Structured control flow (e.g., no GOTOs)
- Callable units (functions / methods / procedures)
- Block structure and scoping

“**Structured program theorem**” gives three building blocks:

- Sequence
- Selection
- Iteration

Type Systems

00110001 ?

$2^5 + 2^4 + 2^0 = 49$?

R of RGB value ?

ASCII Char '1' ?

X86 Opcode XOR ?

The type of a unit of data determines the possible values that data may take on, and the ways it may be operated on.

Ensuring the constraints on types are obeyed is *type checking*:

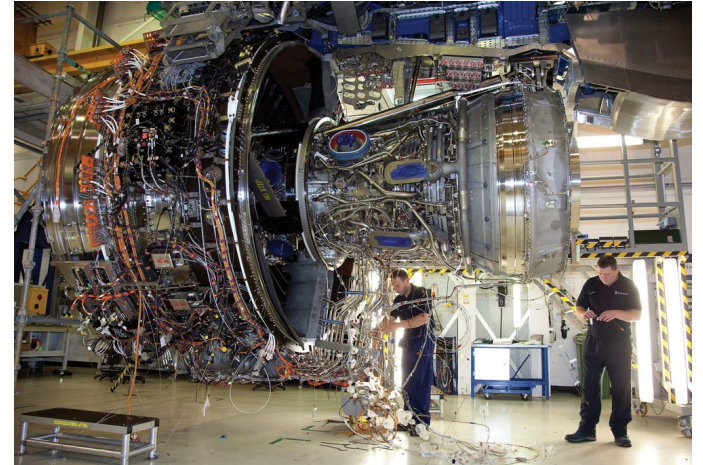
- **Static** type checking: done at **compile time**
 - Java / Haskell / C
- **Dynamic** type checking: done at **runtime**
 - Python / LISP / Javascript

Syntax and Semantics

- **Syntax**: the ways characters can be structured to create a valid program in the given language
 - $3 + 5$: a valid expression involving a number, a binary operator, and then another number
 - *Fair warning*: You will see some syntax that has not yet been explained, especially in the first weeks.
- **Semantics** (meaning / behaviour): what that syntax represents / how the program will behave
 - $3 + 5$: evaluates to a new integer (8) that is the sum of the two integer operands (informally)

Abstraction

- Controlling **complexity**.
- Forming **modules** / components that hide unimportant details and provide an intuitive **interface** to other components.
- Enabling more of the system to fit in our limited fleshy brains at once, without losing the key interactions.
- **Generalising** capability.
- Critical in all languages / paradigms.



Rolls Royce Trent XWB for the A350. Photo: AINonline

The Java Standard Library

- The Java language is augmented with a large standard library (like libstdc++ for C++, .NET for C#, and many others)
 - I/O (accessing files, network, etc)
 - Standard data structures
 - Graphics
 - And much more

<https://docs.oracle.com/en/java/javase/17/docs/api/index.html>

- Learning to use the standard library is part of learning a major language, such as Java.
- Rich and well-designed standard libraries are a key software engineering tool.

The Oracle Java Tutorials

This course follows the structure of the Oracle Java Tutorials for the basic introduction to Java.

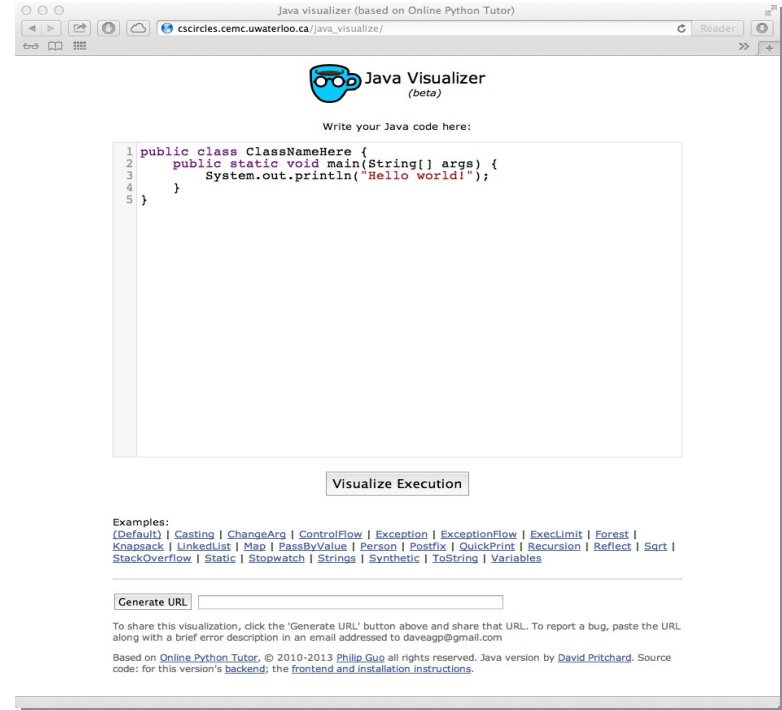
<https://docs.oracle.com/javase/tutorial/>

The tutorials are subject to Oracle's 'Java Tutorial Copyright and License' (Berkeley license).

We will move very fast for the first few weeks. Use the tutorials to ensure that you rapidly become proficient in the basics of Java.

The Waterloo Java Visualiser

Type in simple Java programs and watch step-by-step execution. A great way to enhance your understanding of a new language.



https://cscircles.cemc.uwaterloo.ca/java_visualize/