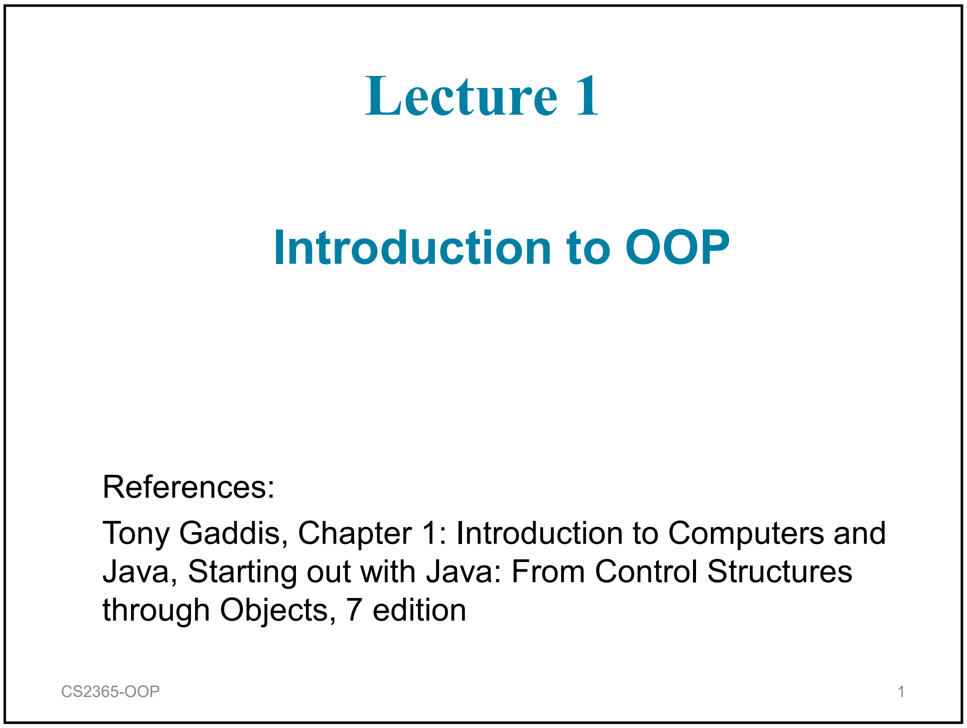
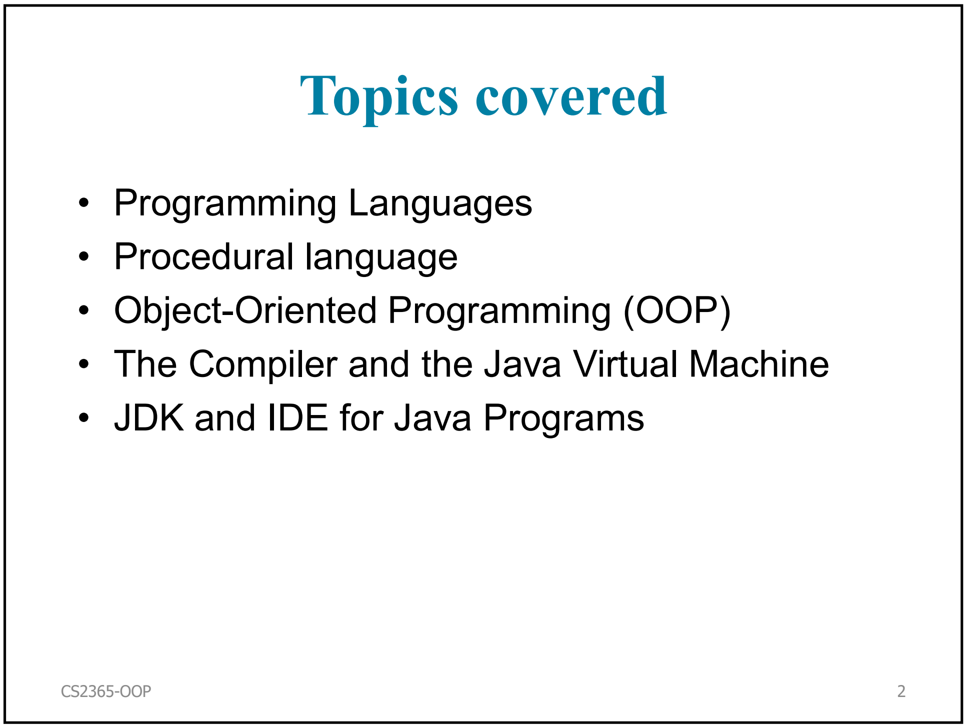


Lecture 1

Introduction to OOP

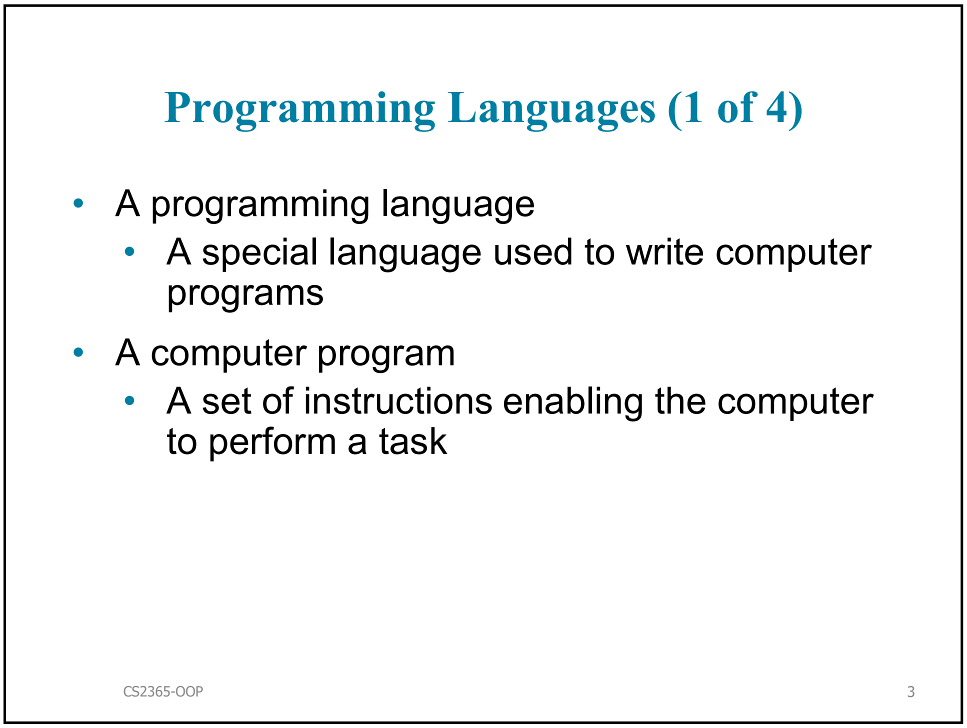
References:

Tony Gaddis, Chapter 1: Introduction to Computers and Java, Starting out with Java: From Control Structures through Objects, 7 edition



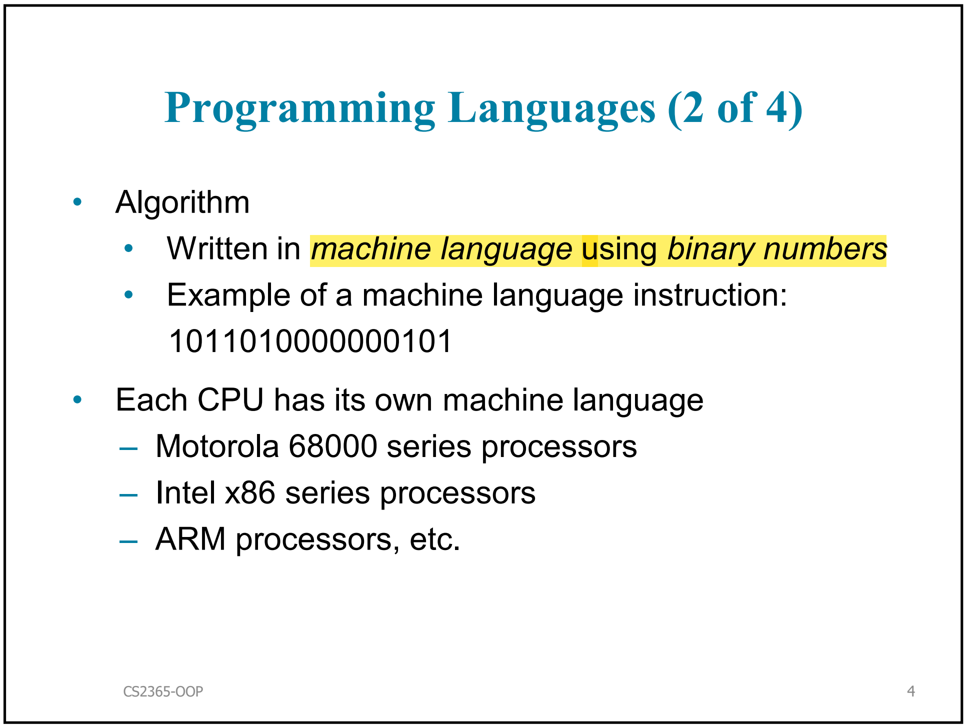
Topics covered

- Programming Languages
- Procedural language
- Object-Oriented Programming (OOP)
- The Compiler and the Java Virtual Machine
- JDK and IDE for Java Programs



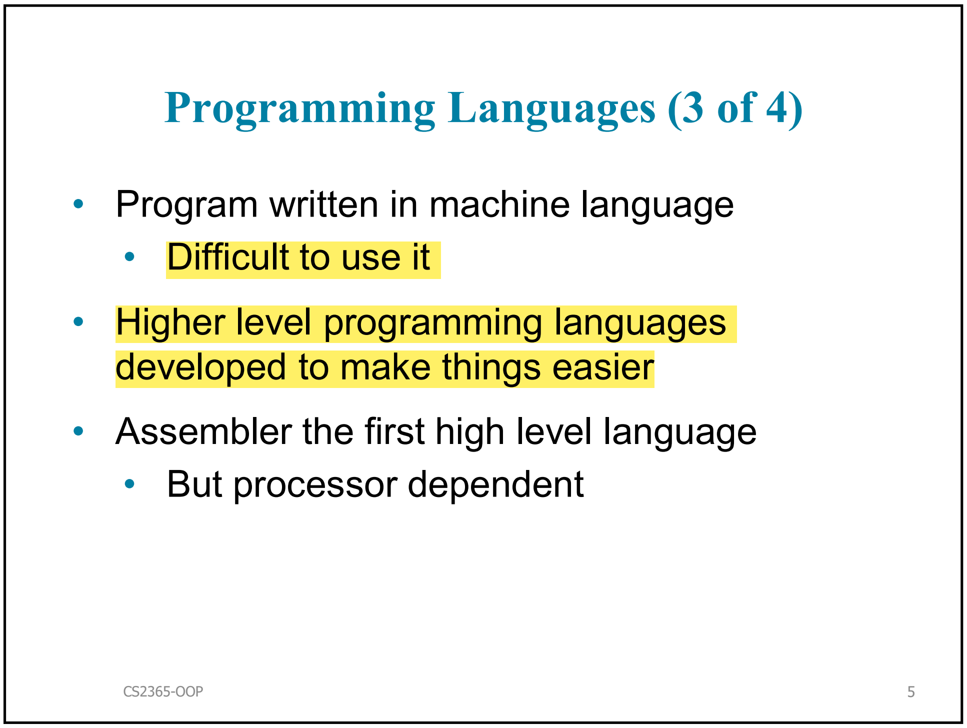
Programming Languages (1 of 4)

- A programming language
  - A special language used to write computer programs
- A computer program
  - A set of instructions enabling the computer to perform a task



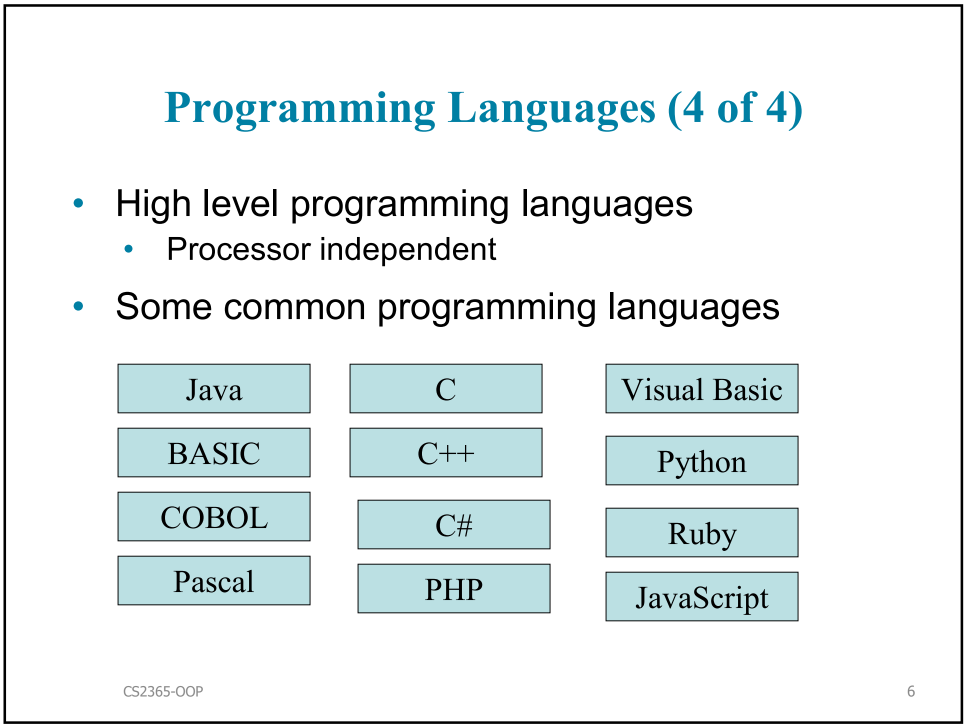
Programming Languages (2 of 4)

- Algorithm
  - Written in *machine language* using *binary numbers*
  - Example of a machine language instruction: 1011010000000101
- Each CPU has its own machine language
  - Motorola 68000 series processors
  - Intel x86 series processors
  - ARM processors, etc.



Programming Languages (3 of 4)

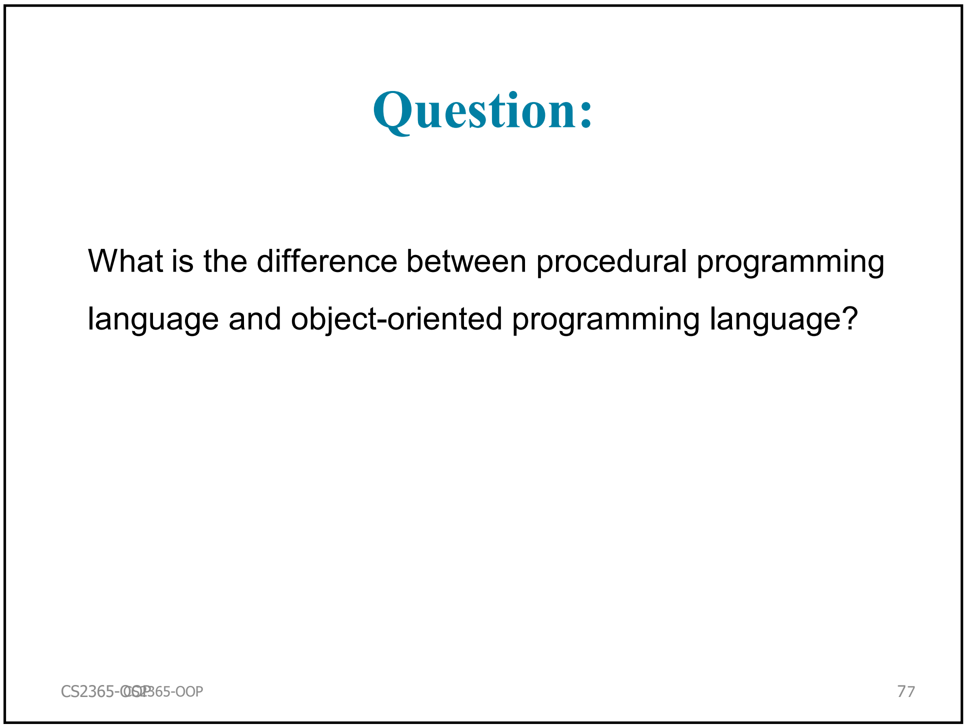
- Program written in machine language
  - Difficult to use it
- Higher level programming languages developed to make things easier
- Assembler the first high level language
  - But processor dependent



Programming Languages (4 of 4)

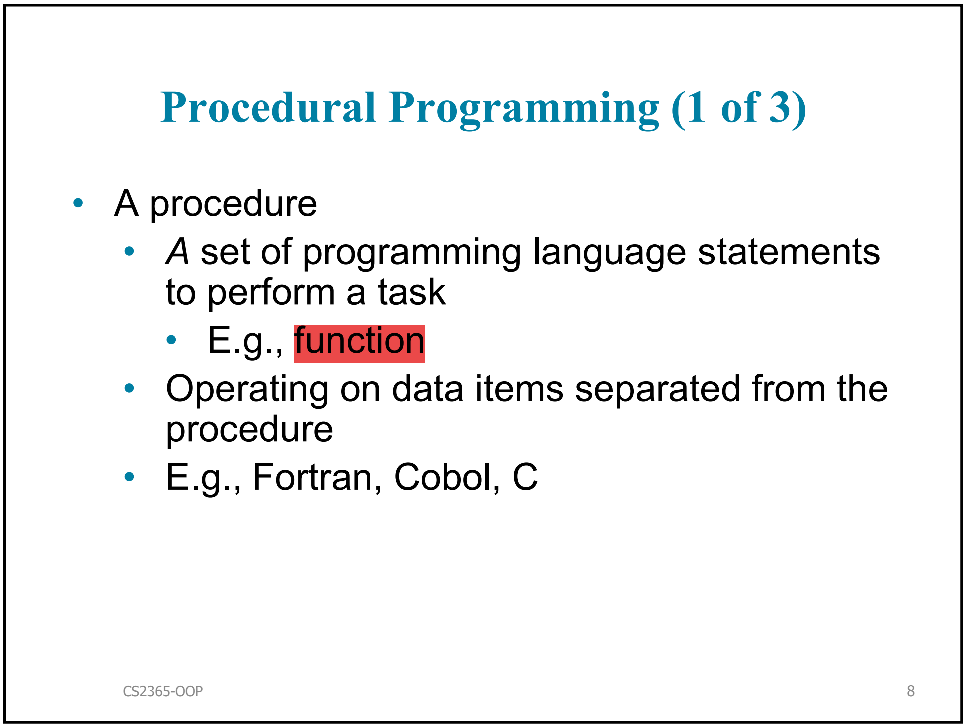
- High level programming languages
  - Processor independent
- Some common programming languages

Java	C	Visual Basic
BASIC	C++	Python
COBOL	C#	Ruby
Pascal	PHP	JavaScript



Question:

What is the difference between procedural programming language and object-oriented programming language?

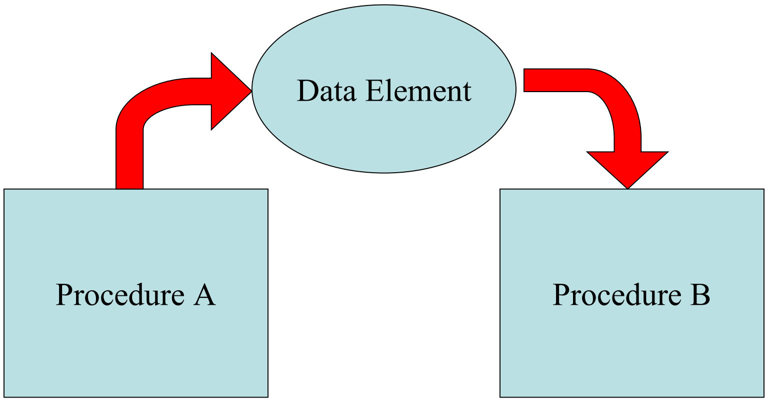


Procedural Programming (1 of 3)

- A procedure
  - *A* set of programming language statements to perform a task
    - E.g., function
  - Operating on data items separated from the procedure
  - E.g., Fortran, Cobol, C

Procedural Programming (2 of 3)

- The data items passed from one procedure to another



Procedural Programming (3 of 3)

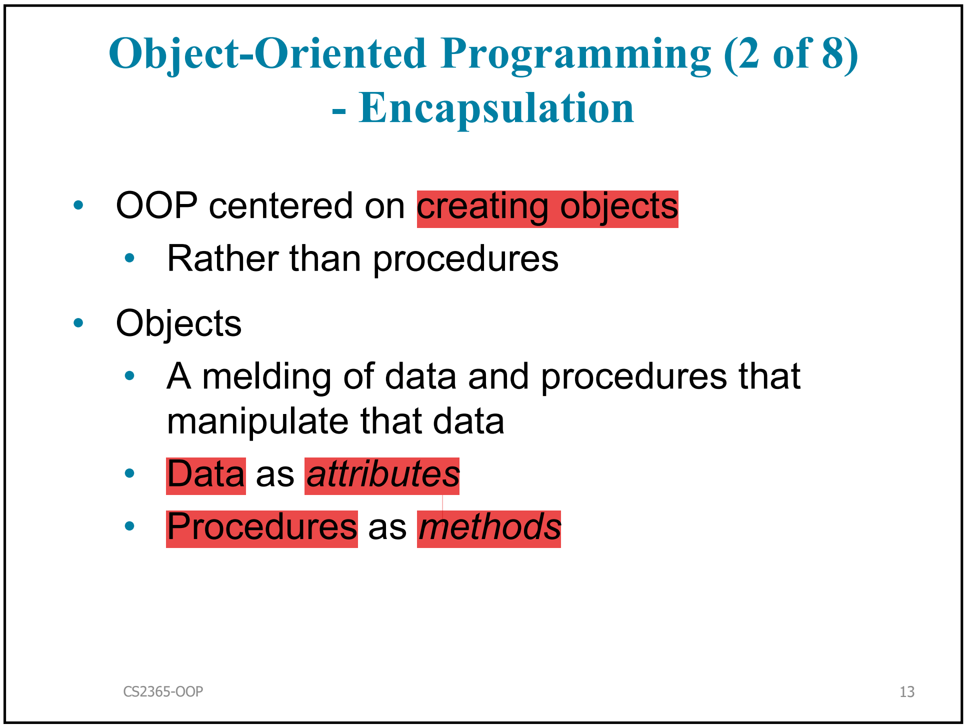
- Data is global to the entire procedures or program
- Data formats might change
  - Thus, the procedures must change

Question:

- Java/C++/C# are an OOP language. Why?

Object-Oriented Programming (1 of 8) - OOP Principles

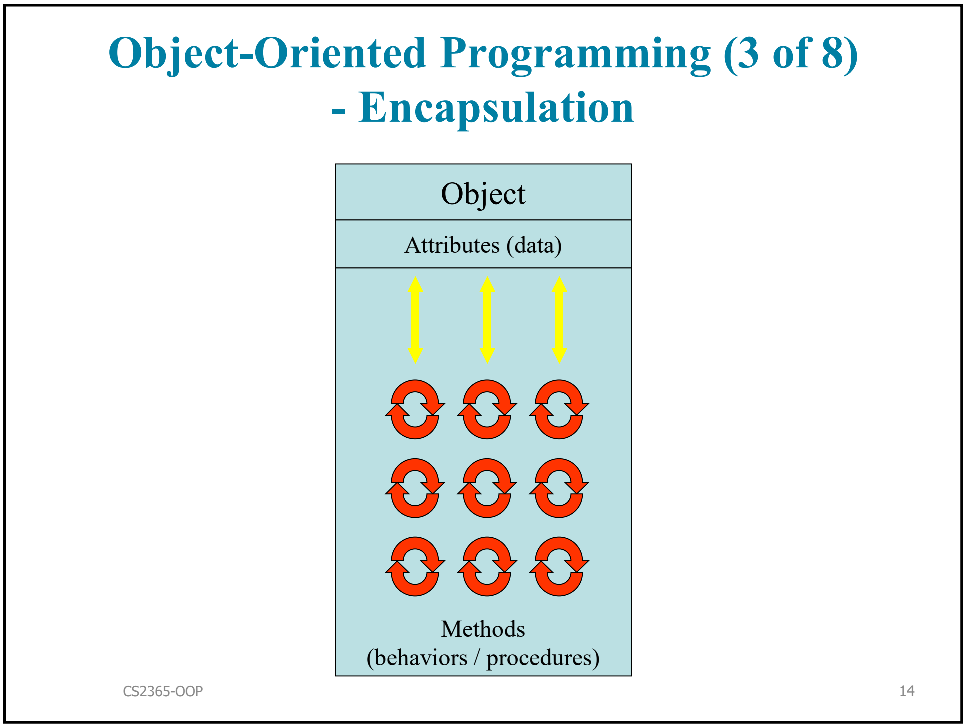
- Encapsulation
  - Data Hiding (information hiding)
- Inheritance
  - Abstraction
- Polymorphism



Object-Oriented Programming (2 of 8) - Encapsulation

- OOP centered on creating objects
  - Rather than procedures
- Objects
  - A melding of data and procedures that manipulate that data
  - Data as *attributes*
  - Procedures as *methods*

13



Object-Oriented Programming (3 of 8) - Encapsulation

Object

Attributes (data)

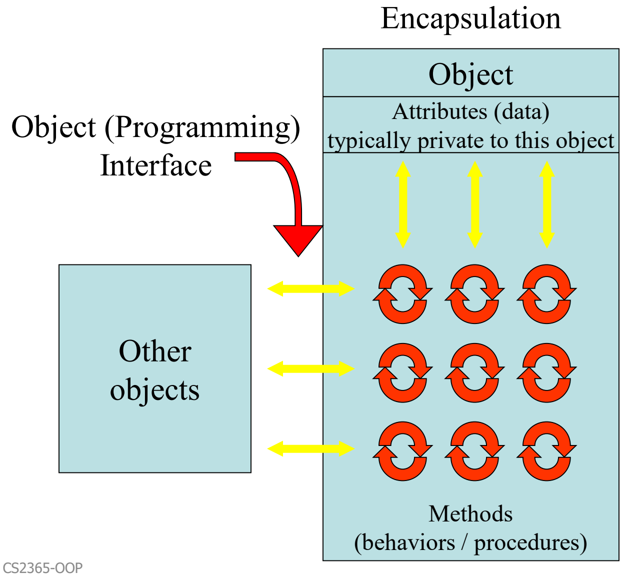
Methods
(behaviors / procedures)

14

Object-Oriented Programming (4 of 8) - Encapsulation

- Encapsulation
  - A mechanism binding data and procedures
  - Object (Class) encapsulating data and procedures
- Data hiding (information hiding)
  - An object hiding data from other objects
- Only an object's methods directly manipulating its attributes
- Other objects indirectly manipulating an object's attributes via the object's methods
  - Known as an object (*programming) interface*

Object-Oriented Programming (5 of 8) - Encapsulation



Object-Oriented Programming (6 of 8) - Inheritance

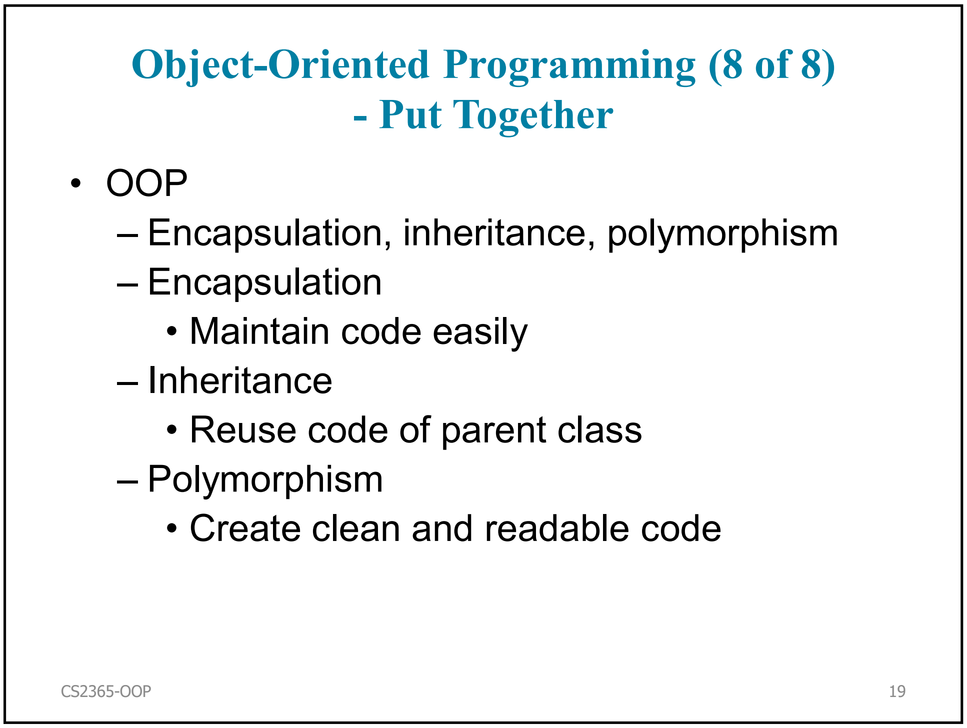
- Inheritance
  - Process acquiring the properties of another object
  - Supports concept of hierarchical classification
    - E.g., Grand parents, Parents, and Me
    - A child inherits its general attributes from its parents
- Class hierarchy abstracted
  - Defer complex details to the next level

17

Object-Oriented Programming (7 of 8) - Polymorphism

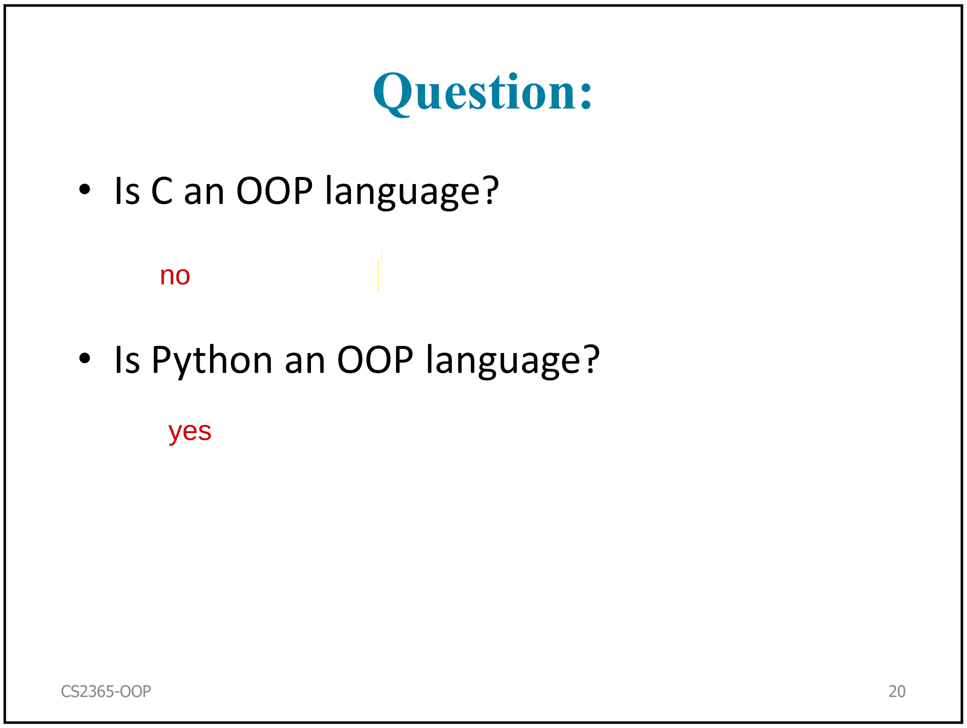
- Polymorphism
  - Meaning “many forms” from Greek
  - Same method name, but different implementations
    - Within a class - overloading
    - On class inheritance - overriding
    - Using interface/abstract class – different implementation

18



Object-Oriented Programming (8 of 8) - Put Together

- OOP
  - Encapsulation, inheritance, polymorphism
  - Encapsulation
    - Maintain code easily
  - Inheritance
    - Reuse code of parent class
  - Polymorphism
    - Create clean and readable code



Question:

- Is C an OOP language?

  no

- Is Python an OOP language?

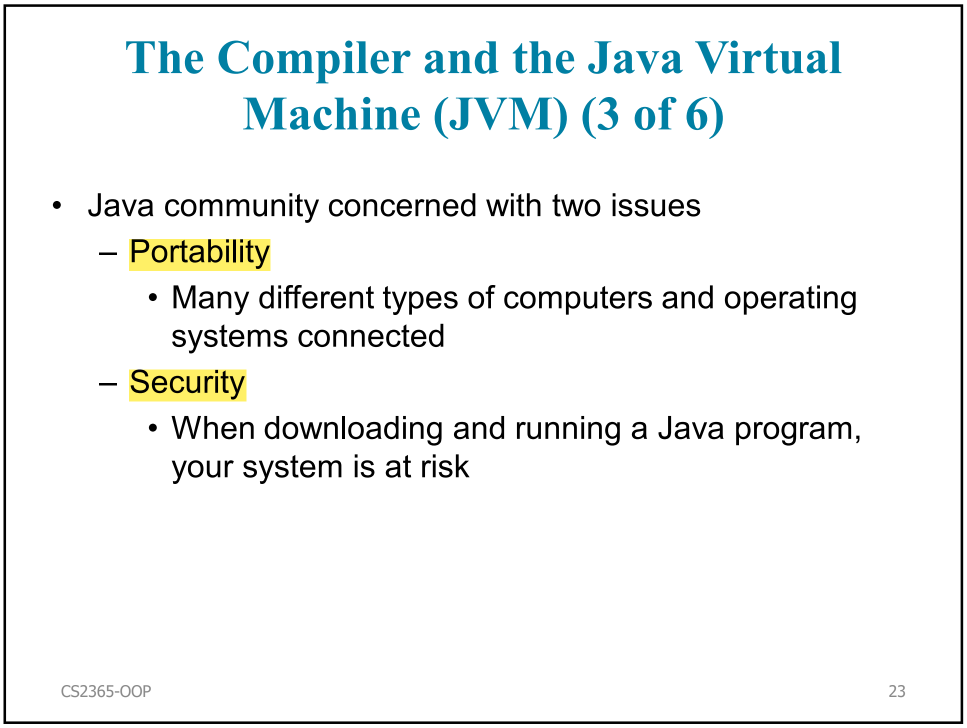
  yes

The Compiler and the Java Virtual Machine (JVM) (1 of 6)

- Java source code files with a *.java* file extension.
- Most compilers
  - Translate source code into *executable* files containing *machine code*
- The Java compiler
  - Translates a Java source file into a file containing *byte code* instructions

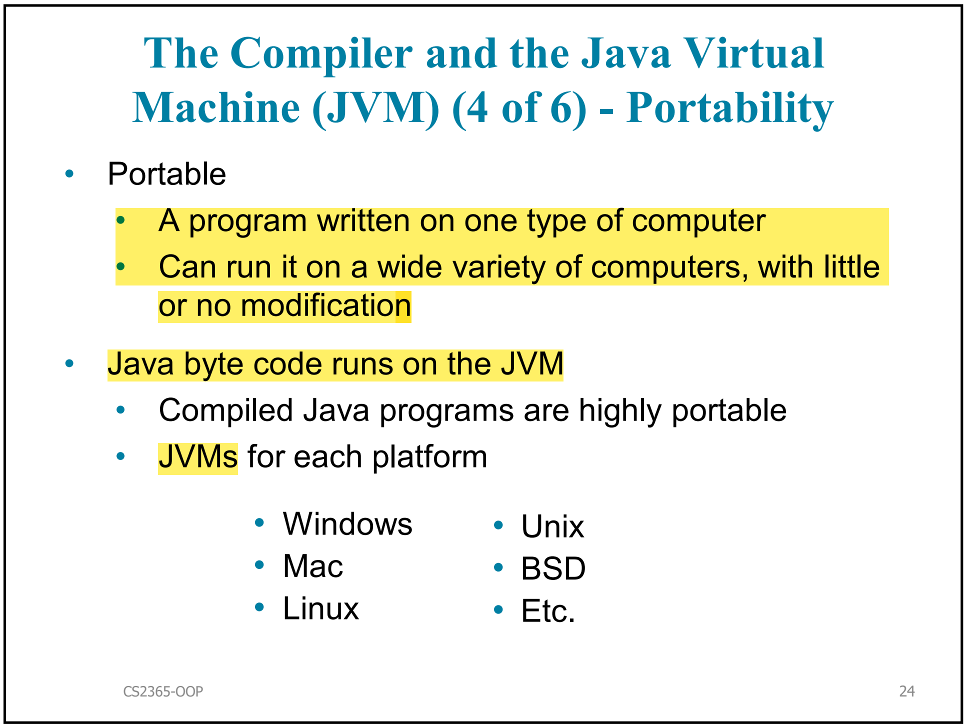
The Compiler and the Java Virtual Machine (2/6)

- Byte code instructions
  - End with the *.class* file extension
  - The machine language
  - But cannot be directly executed by the CPU
- Java Virtual Machine (JVM)
  - A program *emulating* a micro-processor
  - To execute byte code instructions
- JVM called an *interpreter*
  - Java referred to as an *interpreted language*



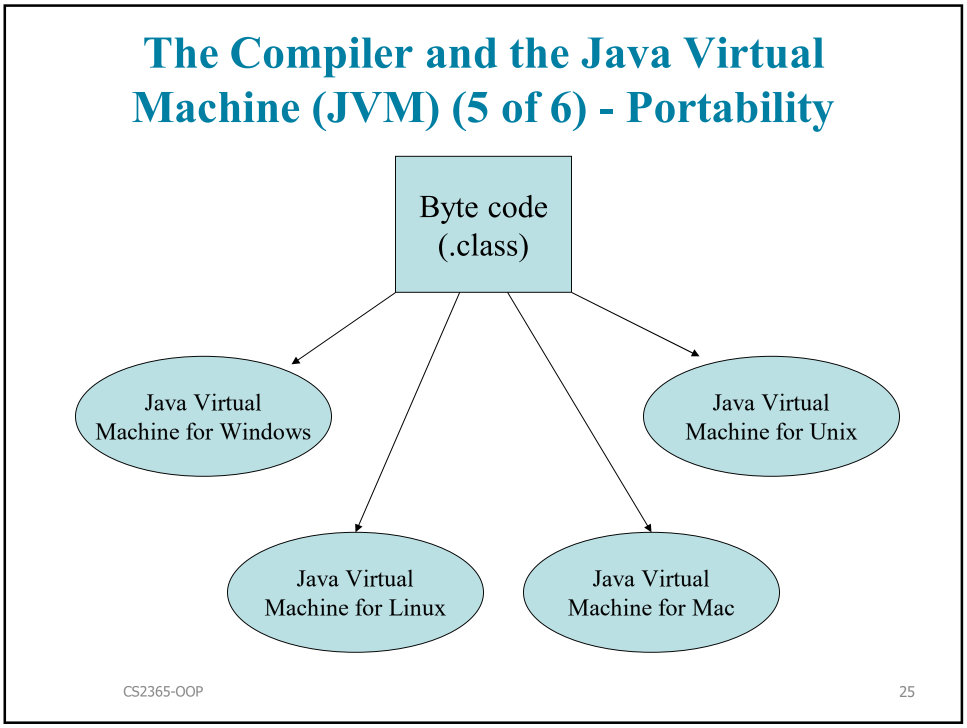
The Compiler and the Java Virtual Machine (JVM) (3 of 6)

- Java community concerned with two issues
  - Portability
    - Many different types of computers and operating systems connected
  - Security
    - When downloading and running a Java program, your system is at risk



The Compiler and the Java Virtual Machine (JVM) (4 of 6) - Portability

- Portable
  - A program written on one type of computer
  - Can run it on a wide variety of computers, with little or no modification
- Java byte code runs on the JVM
  - Compiled Java programs are highly portable
  - JVMs for each platform
    - Windows
    - Mac
    - Linux
    - Unix
    - BSD
    - Etc.



The Compiler and the Java Virtual Machine (JVM) (5 of 6) - Portability

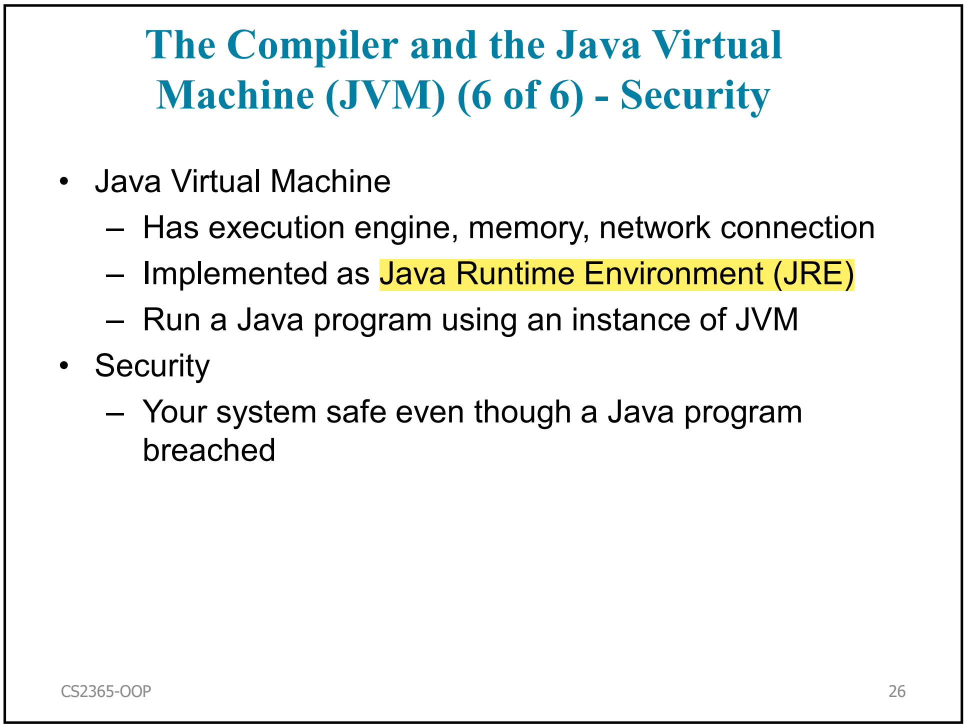
Byte code (.class)

Java Virtual Machine for Windows

Java Virtual Machine for Unix

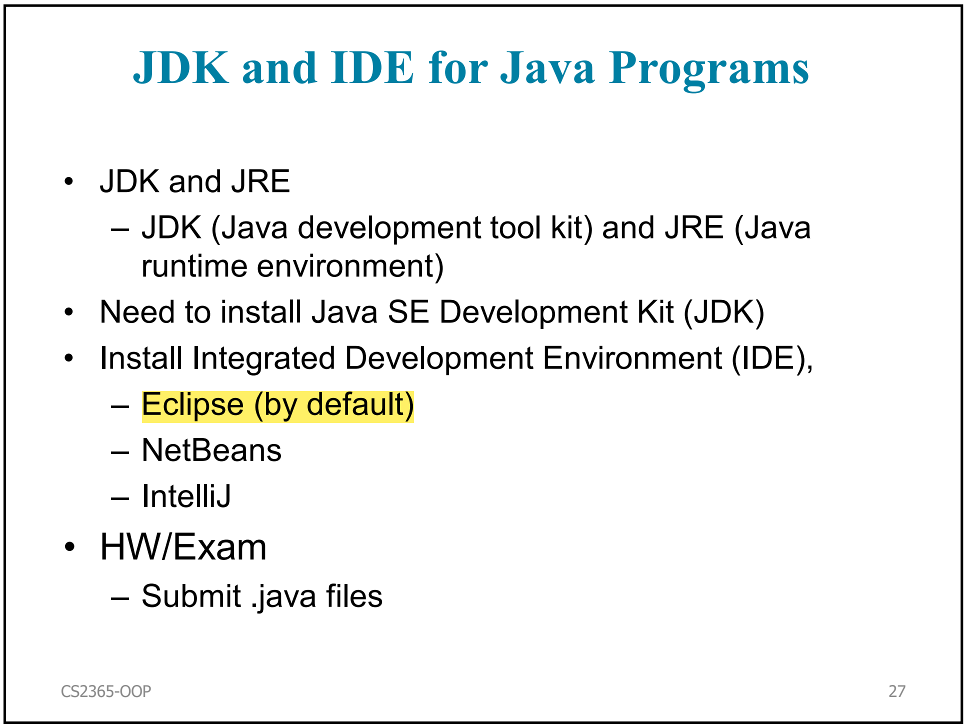
Java Virtual Machine for Linux

Java Virtual Machine for Mac



The Compiler and the Java Virtual Machine (JVM) (6 of 6) - Security

- Java Virtual Machine
  - Has execution engine, memory, network connection
  - Implemented as Java Runtime Environment (JRE)
  - Run a Java program using an instance of JVM
- Security
  - Your system safe even though a Java program breached



JDK and IDE for Java Programs

- JDK and JRE
  - JDK (Java development tool kit) and JRE (Java runtime environment)
- Need to install Java SE Development Kit (JDK)
- Install Integrated Development Environment (IDE),
  - Eclipse (by default)
  - NetBeans
  - IntelliJ
- HW/Exam
  - Submit .java files