Computer Science and Programming I

Garrett Dancik, PhD

https://gdancik.github.io/

Chapter 1: Introduction to Java

Simple View of Running a Program

ex: instructions for computing and printing the sum of two numbers

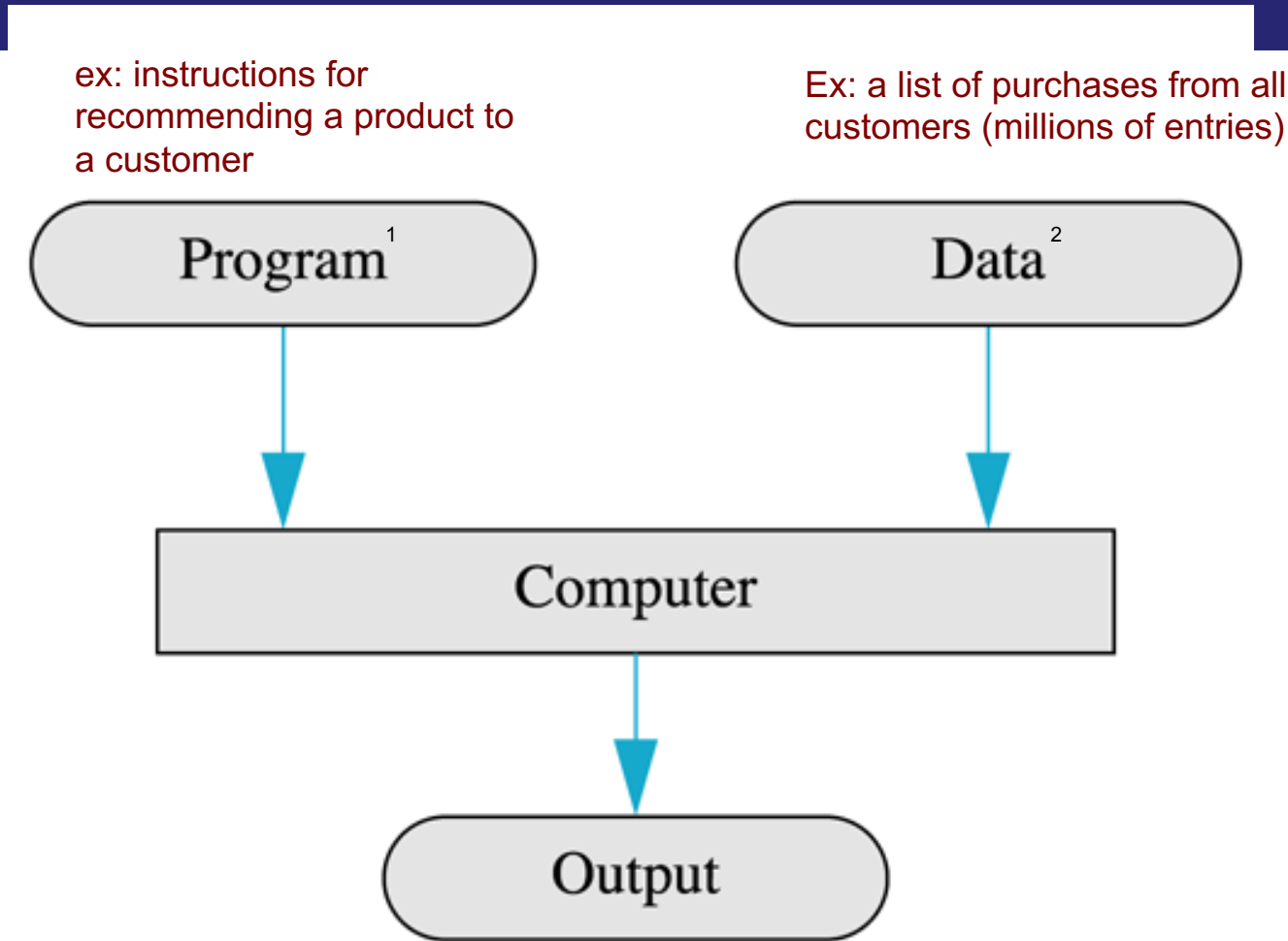ex: the first and second number

Program[1]

Data[2]

Computer

Output
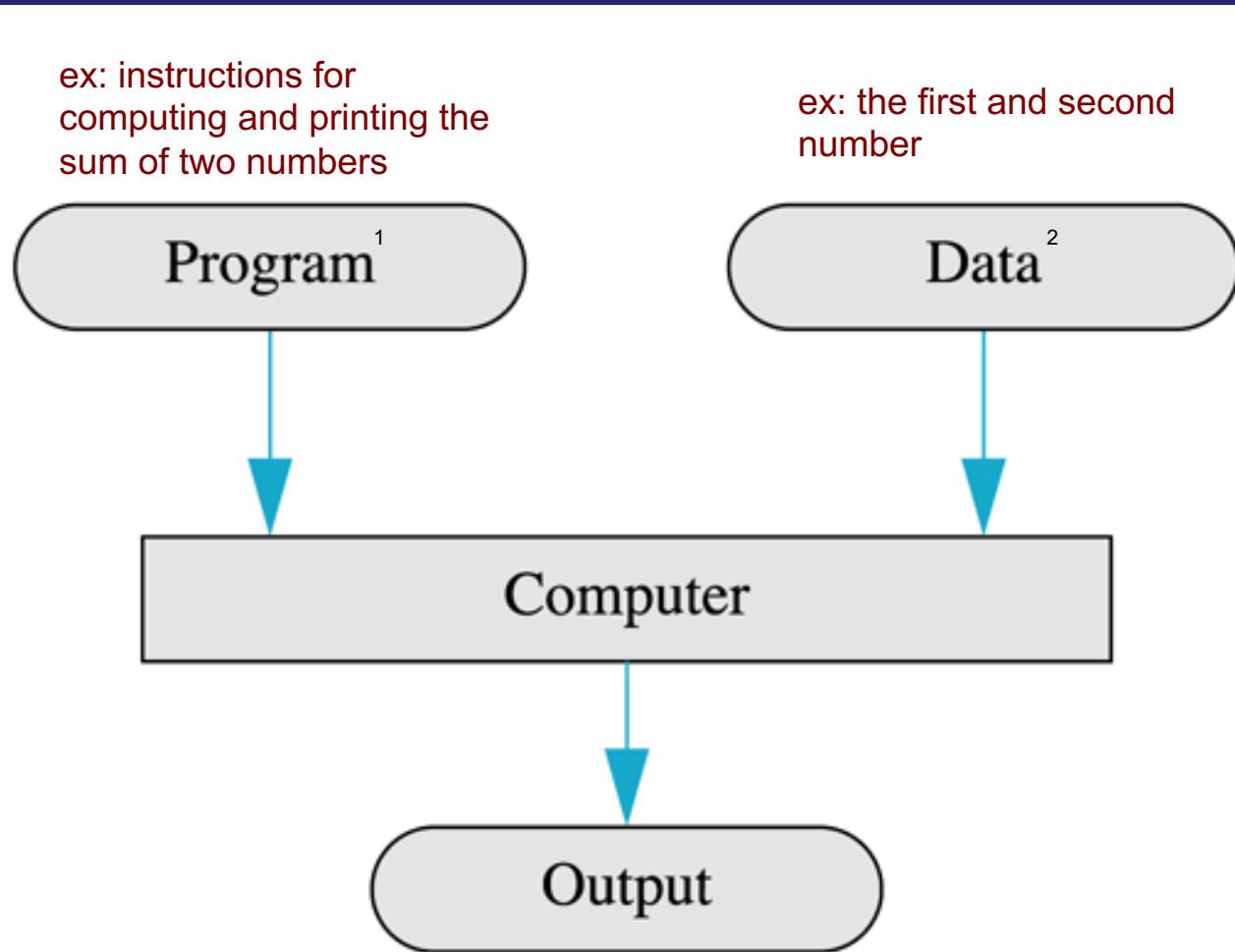
[1]a set of instructions for a computer to follow

ex: the sum of the two numbers

[2]The input into the program

Simple View of Running a Program

ex: instructions for recommending a product to a customer

Ex: a list of purchases from all customers (millions of entries)

Program[1]

Data[2]

Computer

Output

---

[1]a set of instructions for a computer to follow

ex: "featured recommendations"

[2]The input into the program

Algorithms

- Algorithm
  - A sequence of precise instructions that leads to a solution

- Program
  - An algorithm expressed in a language the computer can understand

An Algorithm

Algorithm that determines how many times a name occurs in a list of names:

1. Get the list of names.
2. Get the name being checked.
3. Set a counter to zero.
4. Do the following for each name on the list:
   Compare the name on the list to the name being checked, and if the names are the same, then add one to the counter.
5. Announce that the answer is the number indicated by the counter.

Devise an algorithm for determining the sum of two numbers entered by a user

Let's look at a program that does this!

How does a computer carry out the instructions of a program?

- Computers only understand bits (0s and 1s)
  - Why?
- A computer program, which is generally written in a *high-level language*, must be translated into *machine language*, which consists of 0s and 1s.

High-level Languages

- Common programming languages include …

   C   C++   Objective C   Java   Pascal   Visual Basic   FORTRAN
Perl COBOL   Lisp   Scheme   Ada   C#   Python

- These high-level languages
  - Resemble human languages
  - Are designed to be easy to read and write
  - Use more complicated instructions than the CPU can follow
  - Must be translated to zeros and ones for the CPU to execute a program

Low-level Languages

- In low-level language the semantics of the language follows the architecture of the machine
- An assembly language command such as

$$ADD \quad X \quad Y \quad Z$$

  might mean add the values found at X and Y in memory, and store the result in location Z.

- All languages must be translated to machine language (zeros and ones)
  - Assembly language: ADD   X   Y   Z
 - Machine language: 0110   1001   1010   1011

Compiling and running a Java program

Source code: the original program in a high level language

The java *compiler* translates java code into bytecode, which can be interpreted into *machine code* by the Java Virtual Machine (JVM)

Java bytecode is platform independent, since it can be run on any computer (with a JVM)

C++ programs, for comparison, are compiled directly into machine code (an executable) and these are not platform independent

source: https://www.safaribooksonline.com/
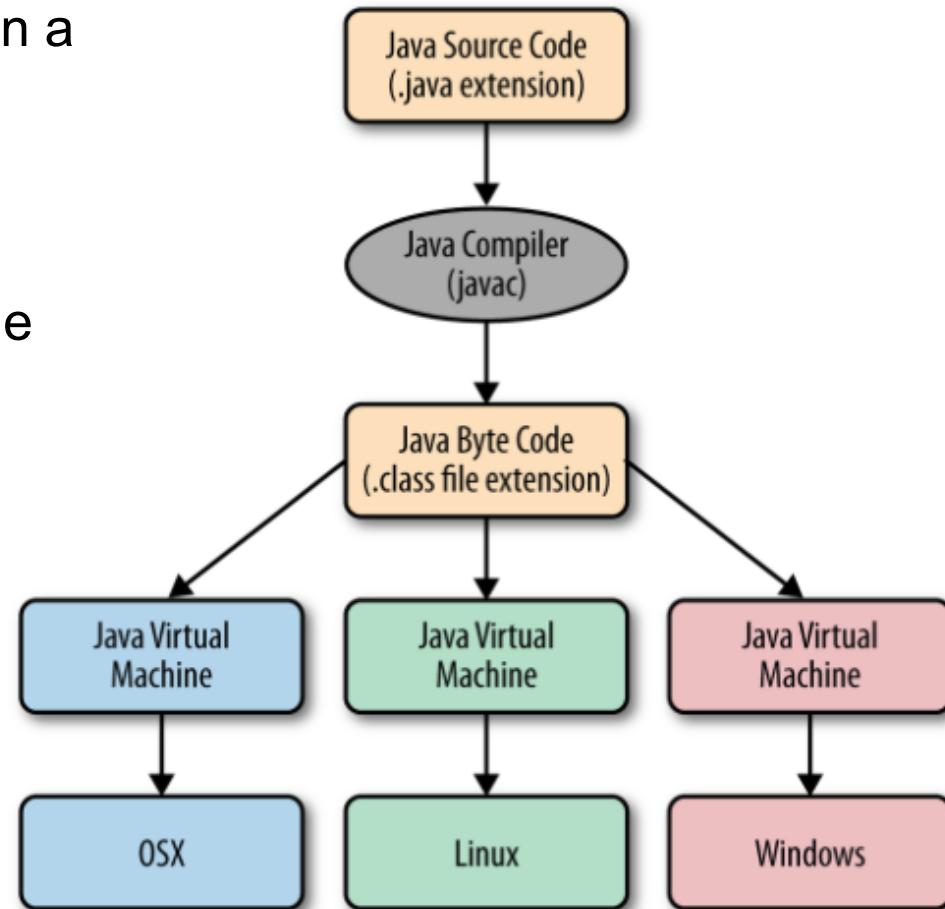
Program Errors

- Syntax errors
  - Violation of the grammar rules of the language
  - Discovered by the java compiler
    - Error messages may not always show correct location of errors
- Run-time errors
  - Error conditions detected by the computer at run-time
- Logic errors
  - Errors in the program's algorithm
  - Most difficult to diagnose
  - Computer does not recognize an error – testing is extremely important