

Introduction

Course Goals

- Learn to program in C++
- Acquire the the basics of object-oriented programming
- Use modern software engineering techniques

Organization

- Communication through the course repository/website
- Brightspace
- Weekly lecture
- Labs
- Microsoft Teams

Lectures

- I will upload the code (and slides) on the website before the class, so that you can directly annotate them

Lectures

- **Please interrupt me at any time to ask questions**

Policy

- You are encouraged to consult with your classmates/friends but collaboration in the assignments is **not allowed**
- You are **not allowed** to copy code online
- You are **not allowed** to use external libraries (except those provided in the assignments)
- We will use plagiarism tools to validate all homework

High-Level Language

- The high-level languages are English-like and easy to learn and program.
- Example:
 - compute the area of a circle with radius 5:
 - `area = 5 * 5 * 3.1415`

Popular High-Level Languages

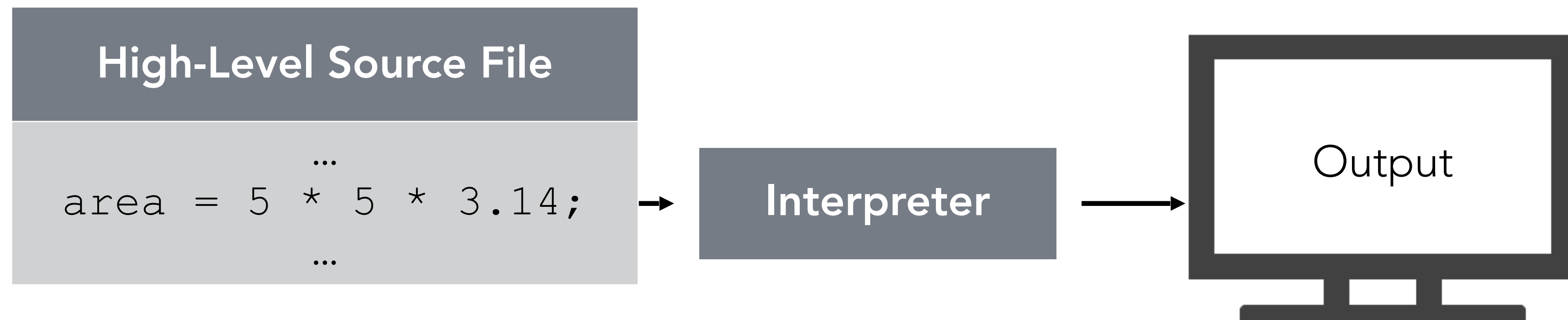
Language	Description
C	Developed at Bell Laboratories. C combines the power of an assembly language with the ease of use and portability of a high-level language.
C++	C++ is an object-oriented language, based on C.
C#	Pronounced "C Sharp". It is a hybrid of Java and C++ and was developed by Microsoft.
Java	Developed by Sun Microsystems, now part of Oracle. It is widely used for developing platform-independent Internet applications.
Python	A simple general-purpose scripting language good for writing short programs.
JavaScript	A high-level, interpreted programming language often used for the Web. It is a language which is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm.
Swift	A general-purpose, multi-paradigm, compiled programming language developed by Apple for iOS, macOS, watchOS, tvOS, and Linux.

Interpreting/Compiling Source Code

- A program written in a high-level language is called a source program or **source code**.
- Because a computer cannot understand a source program, a source program must be translated into machine code for execution.
- The translation can be done using another programming tool called an interpreter or a compiler.

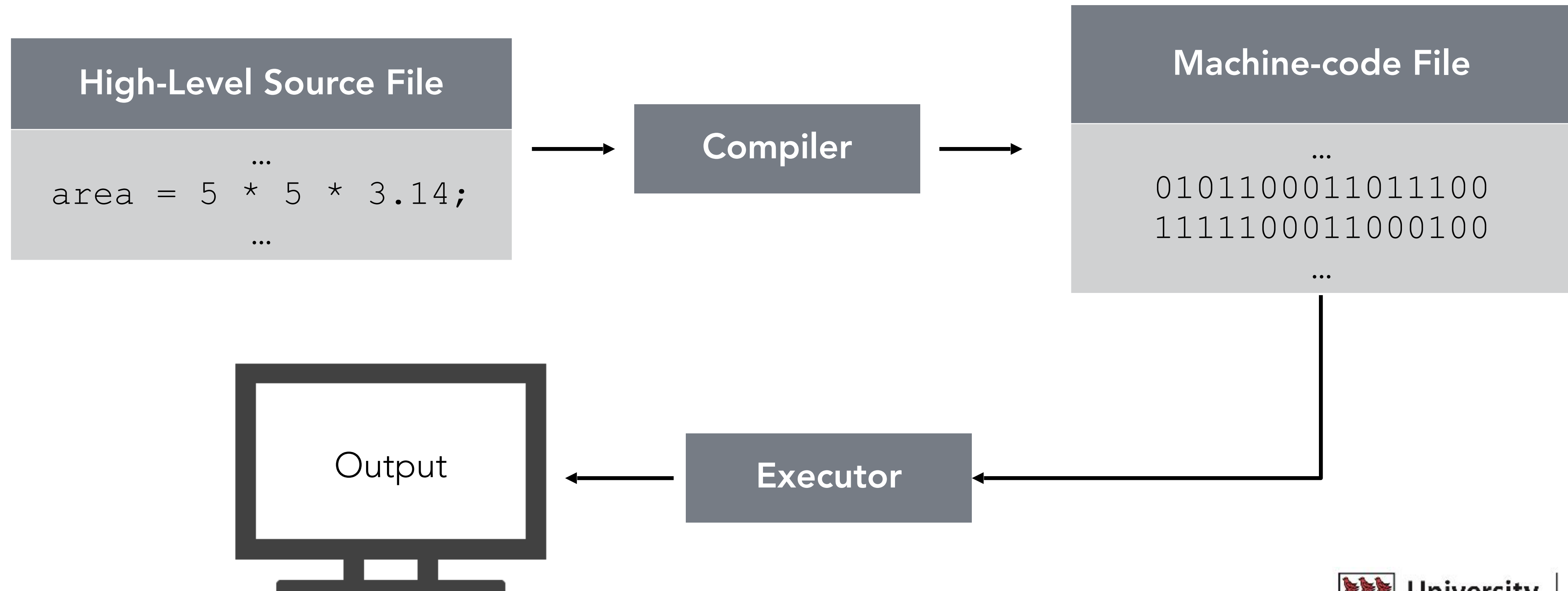
Interpreting Source Code

- An interpreter
 - reads one statement from the source code,
 - translates it to the machine code or virtual machine code,
 - then executes it right away.
- Note that a statement from the source code may be translated into several machine instructions.



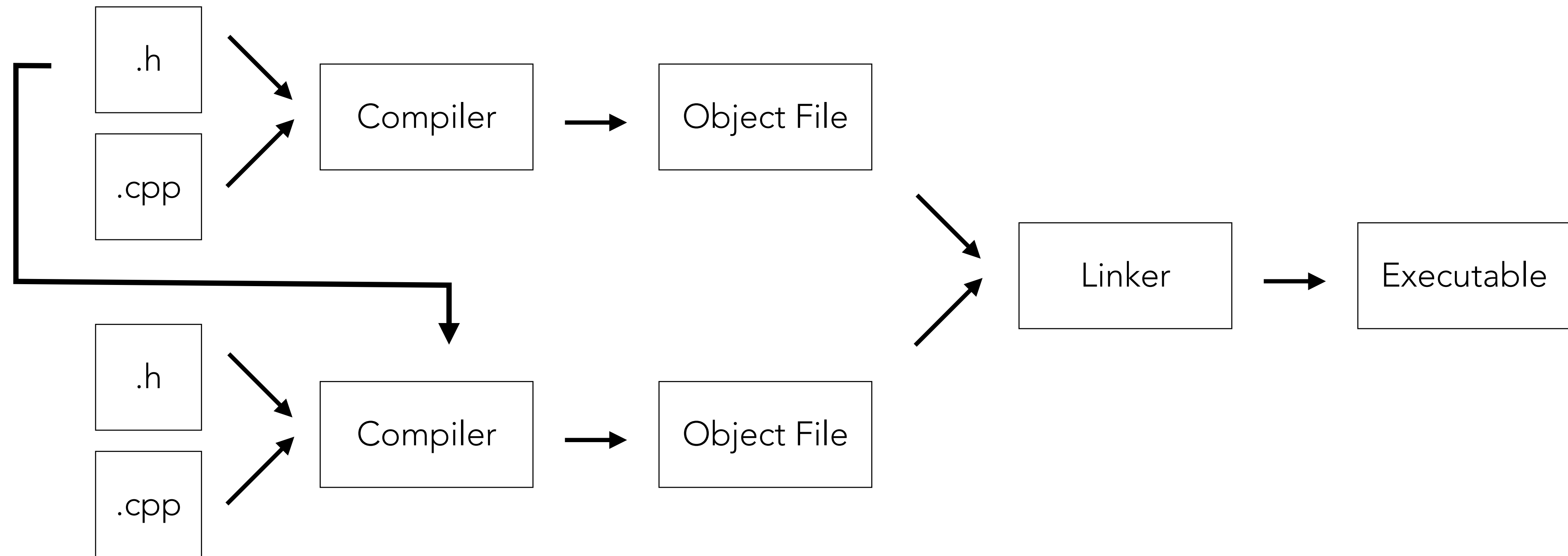
Compiling Source Code

- A compiler translates the entire source code into a machine-code file, and the machine-code file is then executed



What is C++?

- It is a compiled language:



Why C++?

- It allows to write highly efficient code in a convenient way
- I will give an overview of its main features
- The quality of the code will not be evaluated in the assignments.
However, if you learn how to write good C++ code it will greatly simplify the homework

C++

- It is flexible, and many of the features are optional:
 - it can be used as an extension of C, with no objects
 - it can be used as a fully object oriented language
 - it has many advanced features such as “templates” that are very useful to write efficient code that *is also readable*

Comparison with Java

- **Java:** Everything must be placed in a class. **C++:** We can define functions and variables outside a class, using the same syntax used in ANSI C. The "main" function must be defined outside a class.
- **Java:** All user-defined types are classes. **C++:** We can define C types (enum, struct, array).
- **Java:** Single Inheritance. **C++:** Multiple Inheritance

Comparison with Java

- **Java:** No explicit pointers. **C++:** Explicit pointers and “safe pointers” (called Reference) are available.
- **Java:** Automatic memory management. **C++:** Manual memory management (like C) or semi-automatic management (shared pointers, http://en.cppreference.com/w/cpp/memory/shared_ptr).
- **Java:** All objects are allocated on the heap. **C++:** An object can be allocated on the heap or on the stack.



Acknowledgments

Most of the materials in this course are based on Bill Bird CSC116 course