

INTRODUCING...
INTEGRATED DEVELOPMENT ENVIRONMENTS
DR IAN CORNELIUS

HELLO

- Learning Outcomes:
 1. Understanding what an IDE is and its purpose
 2. The ability to use the IDE of choice for this module

INTRODUCTION TO INTEGRATED DEVELOPMENT ENVIRONMENTS

- Integrated Development Environments (IDE) is an application that provides tools for software development
- IDEs generally comprise of:
 - a code editor
 - compiler, interpreter and linker
 - debuggers and graphical user-interface builders

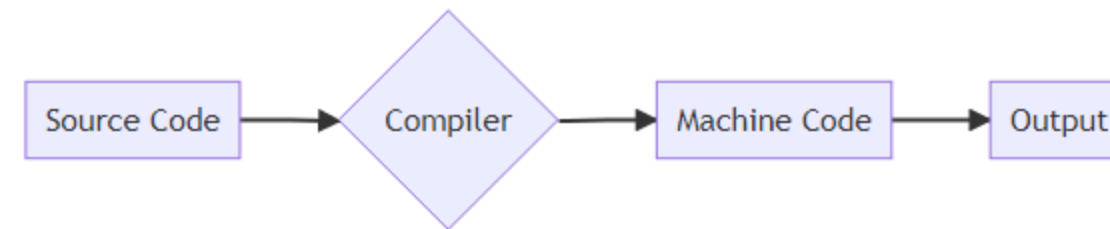
CODE EDITOR

- A text editor program that is designed specifically for editing the source-code of an application
- Can be a stand-alone application, or built into an integrated development environment

COMPILER, INTERPRETER AND LINKER (1)

COMPILER

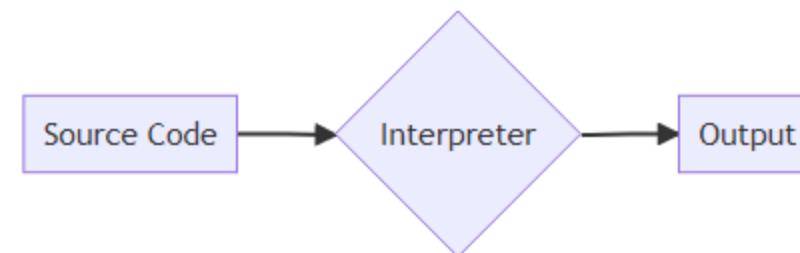
- A tool to transform source-code written in a programming language into object code of the target language
- Simply, converts a program from a human-readable format into a machine-readable format



COMPILER, INTERPRETER AND LINKER (2)

INTERPRETER

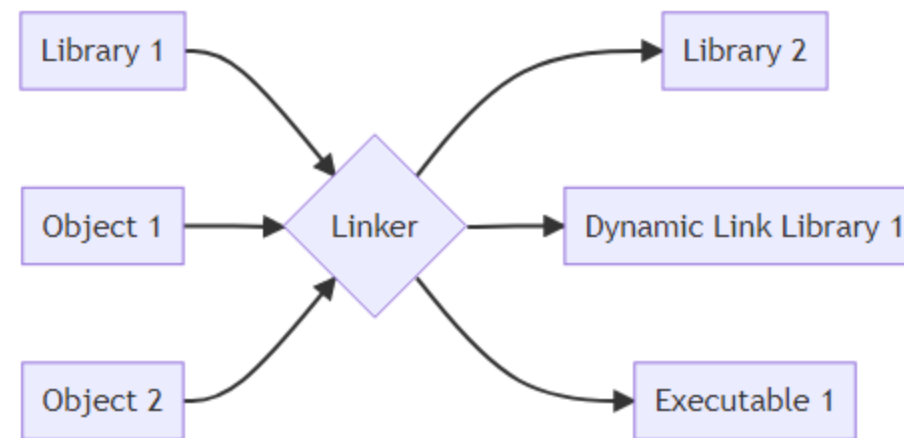
- A tool that executes instructions that are written in a programming language
- Translates high-level instructions into an intermediate form



COMPILER, INTERPRETER AND LINKER (3)

LINKER

- A tool that takes one or more files generated by the compiler and combines them into a single executable file
- Provide links to the libraries needed for the executable program



WHY USE AN IDE? (1)

DEBUGGING

- Instead of using `print()` to debug, use the internal debugger tool
- Enables analysis of source-code line-by-line
 - whilst monitoring and altering variables
 - can watch the output as it is generated
- Removes the tedious-ness of placing multiple `print` statements

CODE REFACTORING AND PROFILING

- Provides the ability to auto-complete code based upon code-intel
- Code refactoring enables you to make global changes, instead of doing it manually
- Code profiling enables you to analyse the code performance on a function-by-function basis
 - i.e. detecting bottlenecks, code completion time

WHY USE AN IDE? (2)

UNIT TESTING

- Automatic test class generation from the source-code
- Code coverage which enables you to analyse the code
 - understand which areas are to be covered by tests and which need more testing
- More information for [creating tests](#)

VERSION CONTROL INTEGRATION

- No requirement to know the commands for each version control system
 - IDE will have tools to commit and push changes to Git
- Enables you to easily keep your project up to date for team members

THE 4061CEM IDE

- Supported IDE: JetBrains IntelliJ IDEA Community/Ultimate
 - [Apply for an Educational License](#)
 - [Download JetBrains IntelliJ IDEA](#)
- Features:
 - Debugging
 - Code Refactoring and Profiling
 - Version Control Integration
 - Python, Java, Kotlin, PHP etc.
- **Note**, you are still expected to learn how to do things via the terminal/command-line

IDE PLUGINS

PYTHON

- Programming language of choice for this course

CODEWITHME

- Enables collaboration between team members
- Sharing code screens to code collaboratively at the same time

IDE INSTALLATION

- Follow one of these guides:
 - [Installing the IDE](#)

IDE DEMONSTRATION

- Demonstration of the IDE
 - Refer to the pre-recorded video for a demonstration



GOODBYE

- Questions?
 - Post them in the **Community Page** on Aula
- Contact Details:
 - Dr Ian Cornelius, ab6459@coventry.ac.uk