Introductory Java, part 2 – J2

- Java packages
- Fully qualified names and import
- The Java standard library

Java packages

- A Java package is a space within which class (and interface) names must be unique.
- The same class name can appear in different packages: for example,
  - `lectures.C01.GCD` and
  - `lectures.J06.GCD`

  are different classes.
- Packages help organise and integrate large code collections.
- Packages can be defined within packages.

- To refer to a class in another package, prepend the package name separated by a dot (“.”):
  - `lectures.C01.GCD`
  - `java.util.Arrays`
- In the same way, to refer to a function in another class, prepend the class name separated by a dot (“.”):
  - `lectures.C01.GCD.gcd`
  - `java.util.Arrays.copyOf`
  - `java.lang.Integer.toString`
- This is the fully qualified name of the class or function/method.

- An `import` declaration makes a class/function from another package available without using its FQN:
  - `import lectures.C01.Factorial;`
  - `import java.util.Arrays;`
  - `import static lectures.C01.Factorial.f;`
  - `import static java.util.Arrays.copyOf;`
- Note: classes in `java.lang` are imported by default.
- Can only import/access classes and functions declared `public`.

The Java standard library

- Java has an extensive *standard library* of ready-made classes providing commonly used functionality.
- For example,
  - `java.lang`: Core Java classes.
  - `java.util`: General utilities.
  - `java.io`, `java.nio`: Input/output.
  - ...and many more.
- `https://docs.oracle.com/en/java/javase/17/docs/api/index.html`