Lecture 3:

C# essentials
Background

C# is Microsoft’s modern OOP language.

- It has many similarities to Java
  - This is by design.
  - C# is intended to be easy for Java programmers to learn.

- C# has some features for lower-level programming, different to Java.
  - They are only used when necessary, and follow C++ to some degree.
  - These are sometimes useful in games for performance or interfacing to low-level APIs.
  - C# is becoming more common in serious game development, Java less so.

- C# is closely tied to Microsoft’s Common Language Runtime (CLR)
Design Goals

The stated design goals for C# (and the CLR) are very similar to Java:

1. A simple, modern general-purpose, object-oriented programming language.

2. Strong type checking, array bounds checking, uninitialized variable detection, portability and garbage collection.
   - Results in software robustness, durability and programmer productivity.

3. Allow software components and distributed environments.

4. Accessible to programmers already familiar with C and C++.

5. Support for internationalization.

6. Support for hosted and embedded systems, ranging from minimal systems to very complex ones.

7. The language is not intended to compete directly on performance and size with C or assembler, but the difference is not a practical concern in most cases.
C# vs Java: Similarities and Differences.

Both are OO languages with a similar syntax for classes, derived from C++.

Both include garbage collection and always use references to objects (unlike C++).

C# allows programmers to define their own value types via the `struct` keyword.

– These can be passed by value, unlike objects.

C# includes restricted pointers – addresses that allow accessing memory directly.

– The restriction is that only code marked as unsafe can use pointers.

C# has unsigned integer types

– This supports low-level coding, such as treating bytes as 0-255.

Both allow automatic boxing and unboxing to convert between primitive types (like `int`) and object types.
C# vs Java: Similarities and Differences.

C# arrays are directly a subclass of the C# collection classes.

C# has multi-dimensional arrays.

C# has properties - accessed like public instance variables, except that they can be implemented with get and set methods.

Both have inner classes, C# is a less restrictive, and has closures.

C# also has delegates, which are roughly lists of pointers to methods of objects.

  – These are often use to register listeners that need to be called when an event occurs.

Both support generics, but:

  – C# generics are processed at compile-time, Java’s at run-time.