C# SCRIPTING AND UNITY BASICS

WHY SCRIPTING ?

It lets you as a programmer, build your own custom logic and gameplay.

VARIABLES

- Variables are containers of data.
- Each variable has its own type/size in C#.(Eg. int is 32 bits in size)
- Each variable has its own name (cannot start with a number, cannot contain any signs except the $ and _, cannot be a keyword)
- Variables are useful for storing necessary data and are a staple of every programming language

```
Eg: int speed = 5;
    float velocity = 15.2f;
    bool isJumping = true;
```

KEYWORDS

- Keywords are the words which provide special meaning to the compiler.
- You cannot name your variables with keywords.

Eg: int,if,for etc.

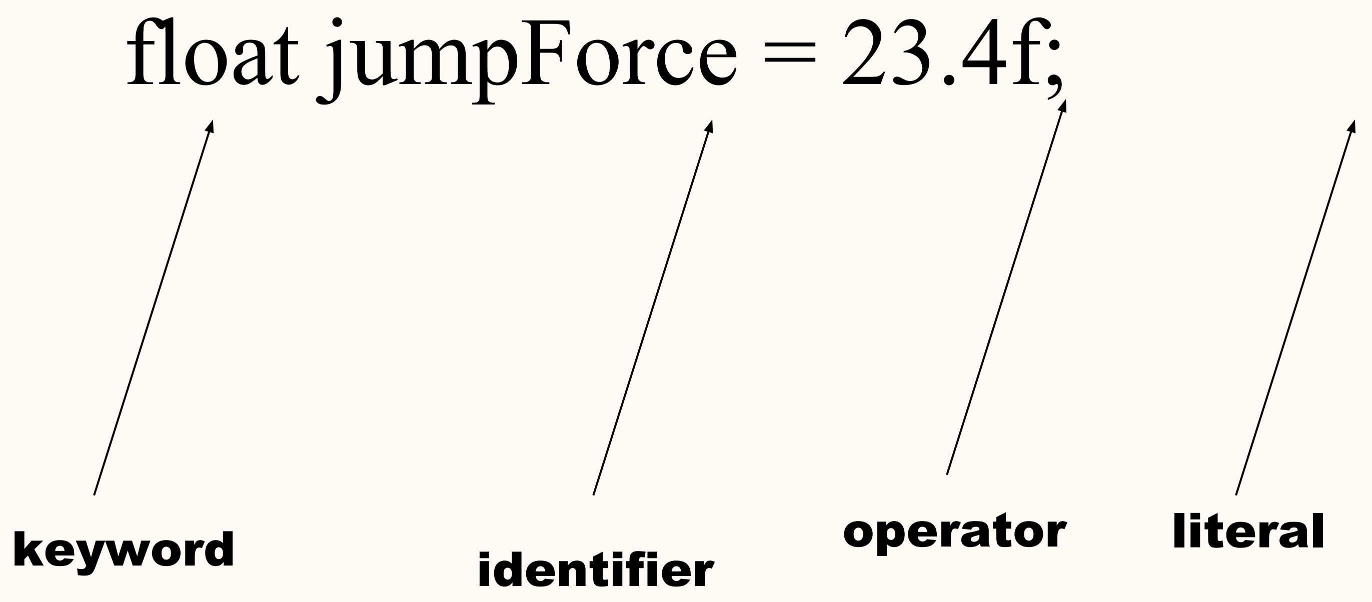
IDENTIFIERS

- identifiers are names given to various program elements, such as variables, functions, arrays.
- Rules for defining valid identifiers are same as the rules mentioned for naming variables.

LITERALS

- literals are fixed values directly embedded within the source code of a program. They represent constant data that does not change during program execution. Literals are used to initialize variables

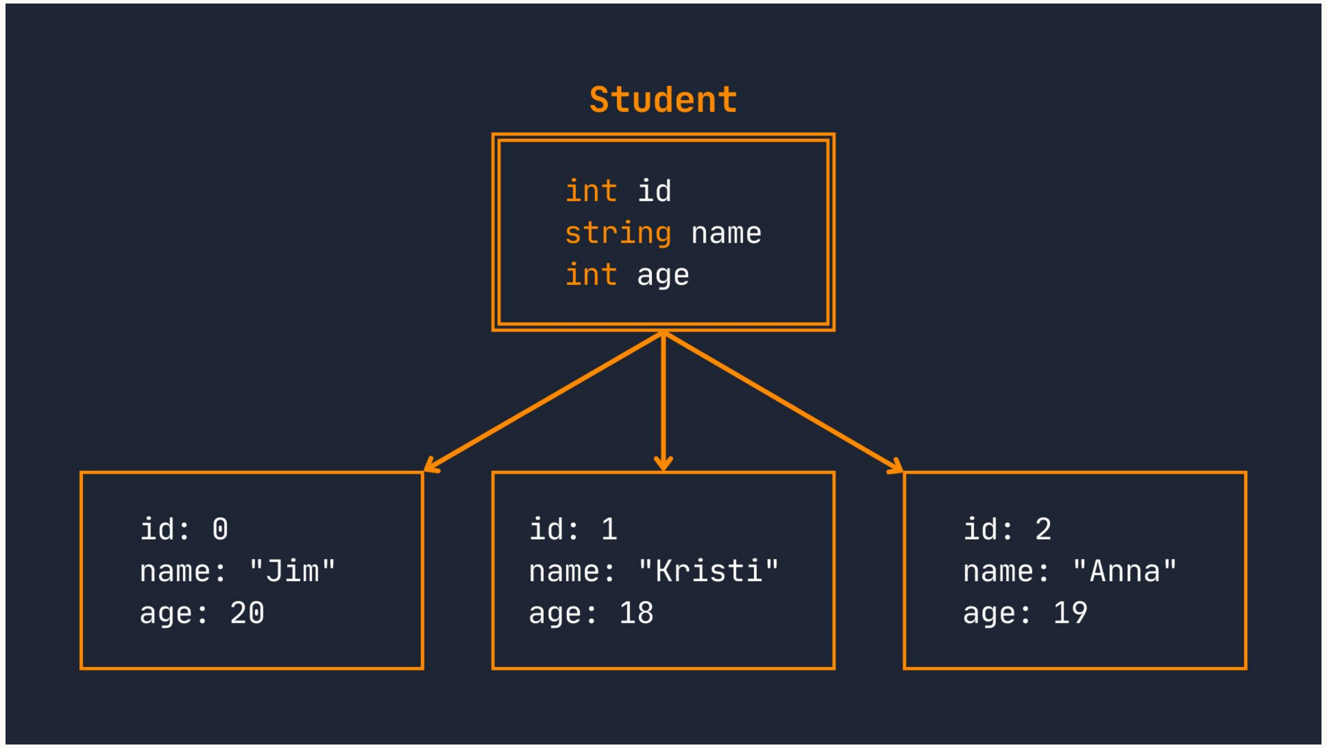
Eg: 1.25f,true,false,23 etc.



```
float jumpForce = 23.4f;
```

OBJECT ORIENTED PROGRAMMING

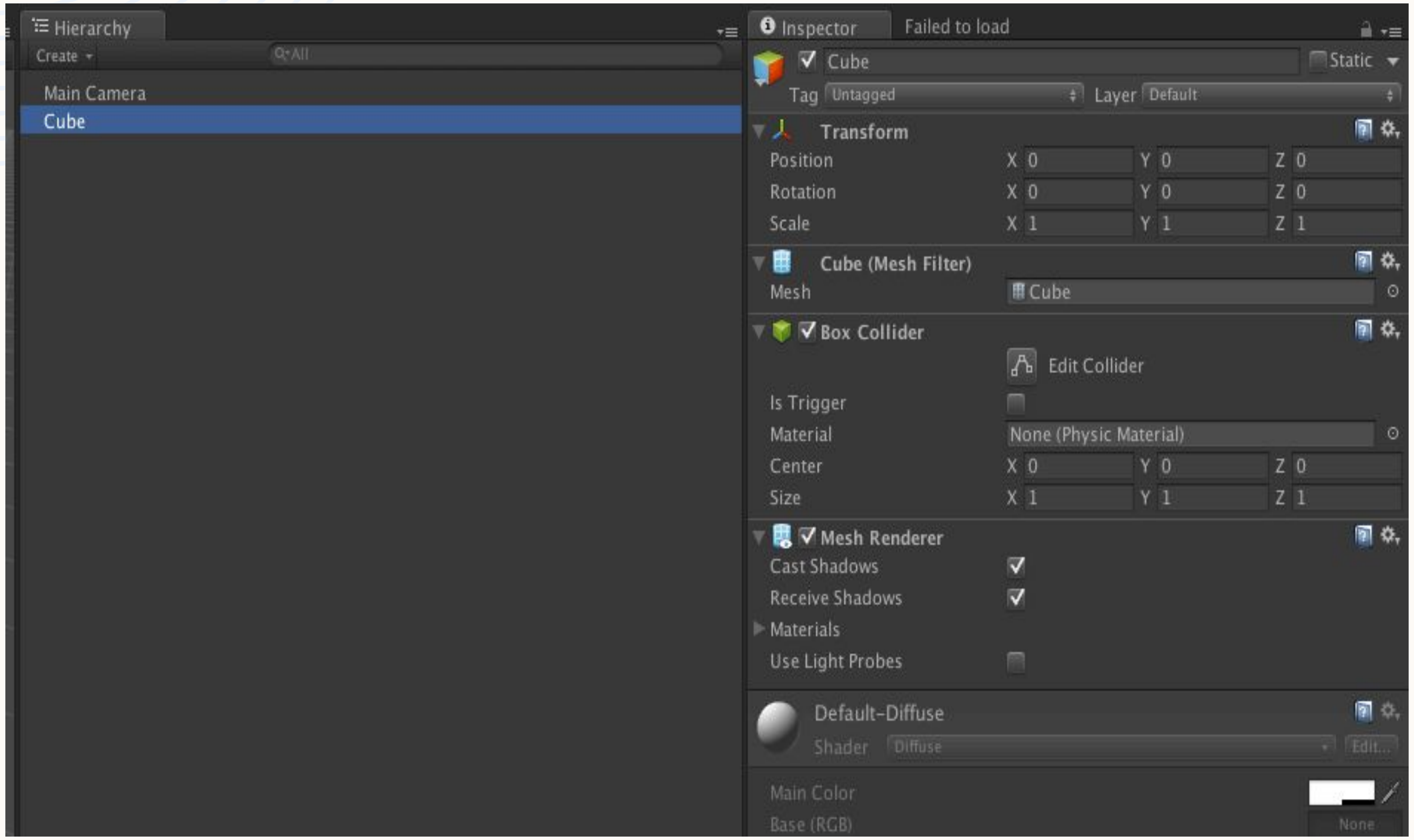
Object oriented programming is a paradigm which organizes your code in self-contained units called objects.



“every object is an instance of a class” and “every class is a blueprint/template of an object”

CLASSES AND OBJECTS

- Classes are user defined data types which are used to defined complex data entities than what our traditional datatypes(primitive data types) such as int can offer.
- And every object is a variable of the data type of class.
- Classes contain data as well as methods which will be useful to operate on that data (this is also called encapsulation).
- Most of the unity components are classes, some examples are Transform, SpriteRenderer, RigidBody etc.



Cube ✓ Static
Tag Untagged Layer Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X 0	Y 0	Z 0
Scale	X 1	Y 1	Z 1

Cube (Mesh Filter)
Mesh Cube

✓ Box Collider
Edit Collider
Is Trigger ☐
Material None (Physic Material)

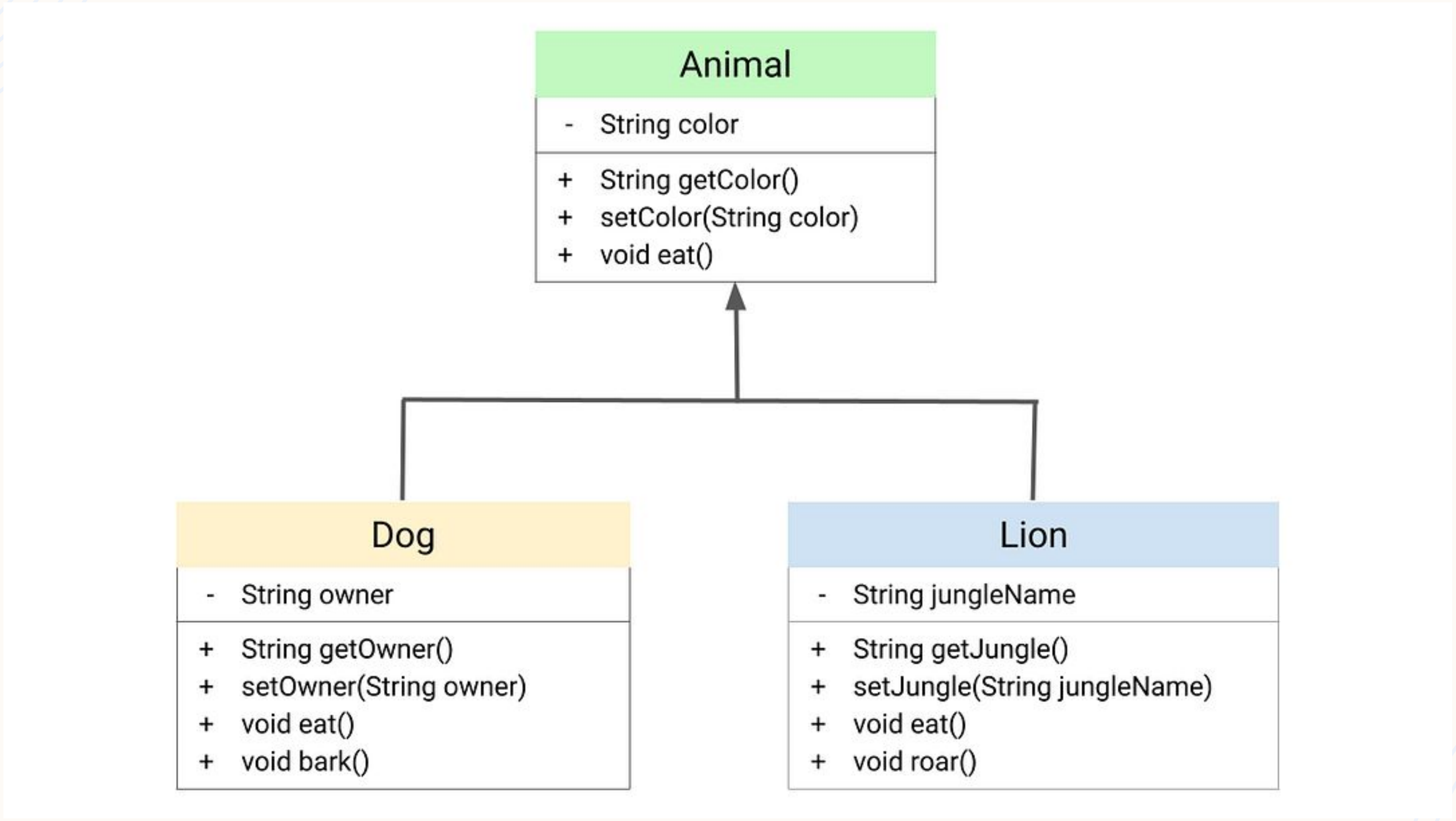
Center	X 0	Y 0	Z 0
Size	X 1	Y 1	Z 1

✓ Mesh Renderer
Cast Shadows ✓
Receive Shadows ✓
Materials
Use Light Probes ☐

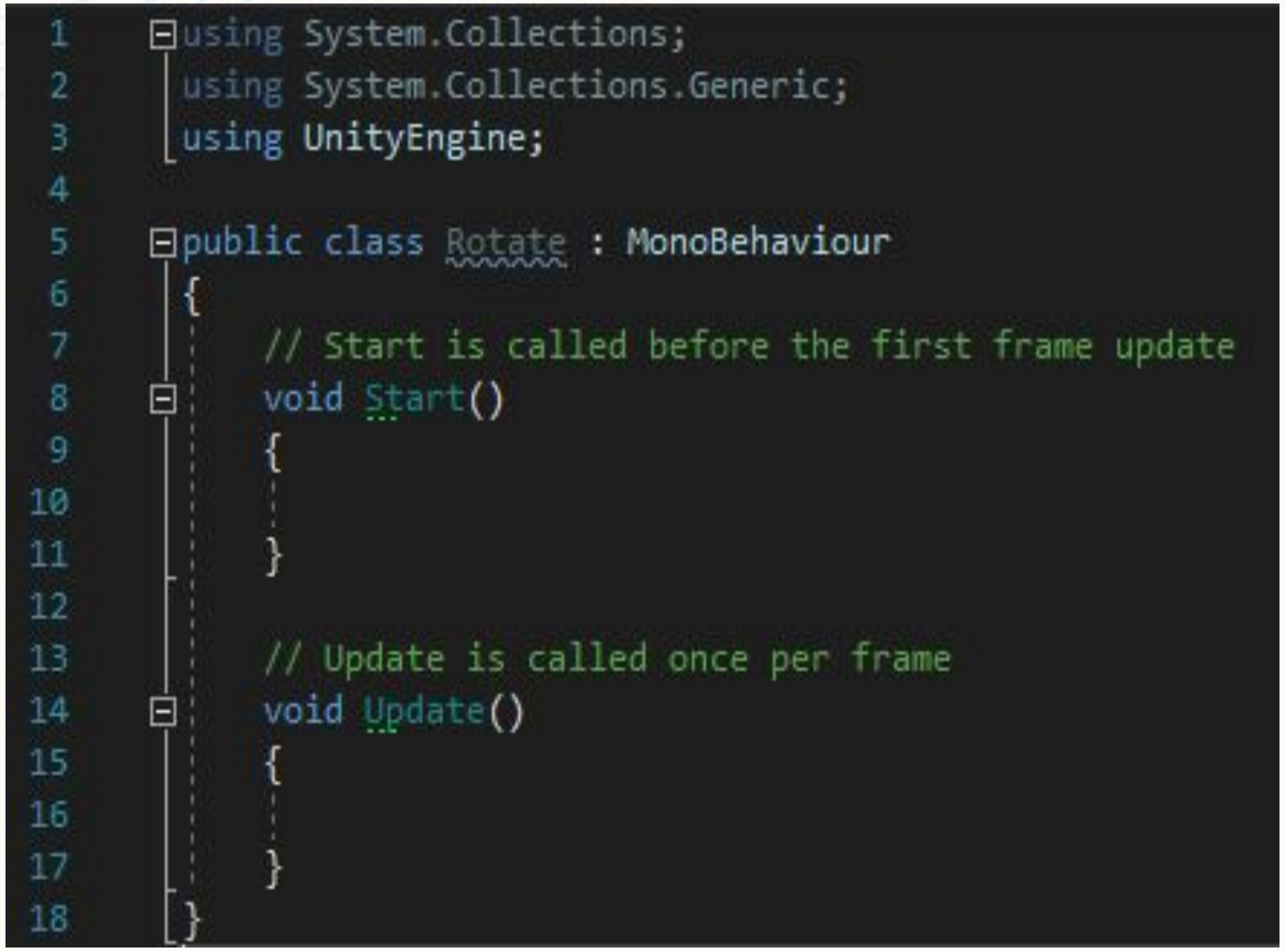
Default-Diffuse
Shader Diffuse Edit...
Main Color
Base (RGB) None

The scripts we will write, will be attached to game objects just like components, and these scripts will be a class too!!

INHERITANCE



INHERITANCE



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rotate : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

- We use ‘:’ to inherit properties from a class.
- You inherit properties from Monobehaviour to your class.
- All Monobehaviour scripts should be attached to gameobjects if they want to be run.

FUNCTION OVERLOADING

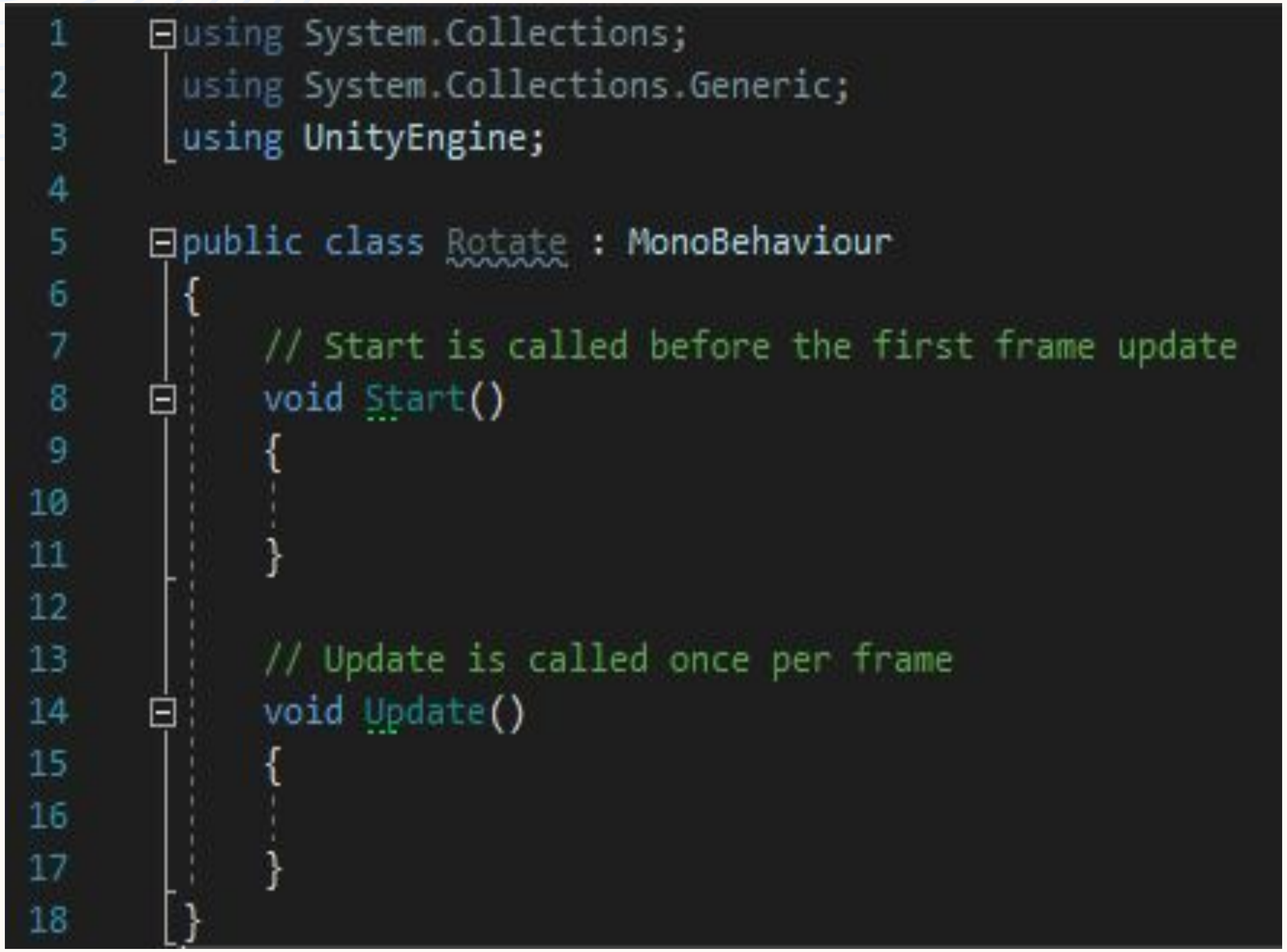
```
int myMethod(int x)

float myMethod(float x)

double myMethod(double x, double y)
```

- Same method names but different parameters and return types
- In Unity you will find many methods from classes which can take a lot of variety of parameters for the convenience of game developers.
- For example: the transform.translate() function has different signatures
- public void **Translate**(Vector3 **translation**);
- public void **Translate**(float **x**, float **y**, float **z**);

START() AND UPDATE()



```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Rotate : MonoBehaviour
{
    // Start is called before the first frame update
    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {

    }
}
```

Start():runs only the first frame. It is useful for initializations you want to make when the game starts.
For eg: setting the initial position of the player in the world

Update():runs one time each frame of the game, useful for repetitive elements in the game loop.
for eg: input system which continuously checks for a spacebar press every frame to perform a jump

LETS GO TO THE UNITY EDITOR NOW!!