Object Orientation

Software Engineering 1

Dr Swen E. Gaudl

University of Plymouth 2022

COMP10000

Prep For Thursday:

- Finish Exercise3
- Come with questions for the seminar!
- Work through the exercise during the session to get help.
- If you need help with underlying fundamental concepts contact PALS!

COMP1000 Agenda This Week:

- Topics from Thursday
- OOD
- Classes and Program Structure
- Q&A

Topics:

- What are Loops?
  - https://www.tutorialspoint.com/csharp/csharp_loops.htm

```
Console.WriteLine("Counting Up: " + 0);
Console.WriteLine("Counting Up: " + 1);
Console.WriteLine("Counting Up: " + 2);
Console.WriteLine("Counting Up: " + 3);
Console.WriteLine("Counting Up: " + 4);
```

Topics:

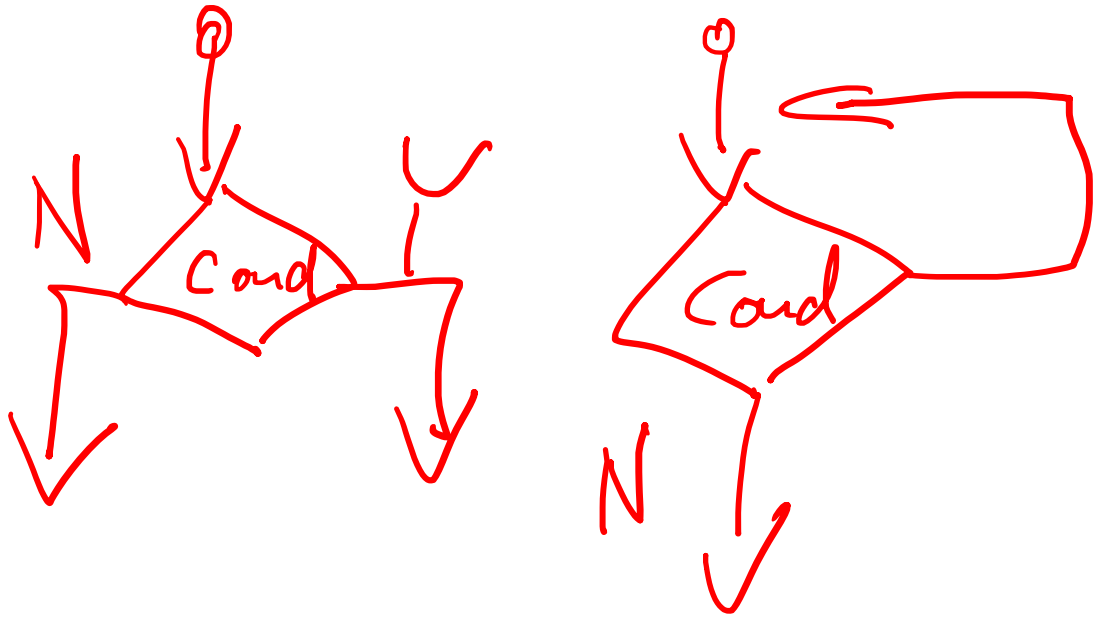
- What are Loops?
  - https://www.tutorialspoint.com/csharp/csharp_loops.htm

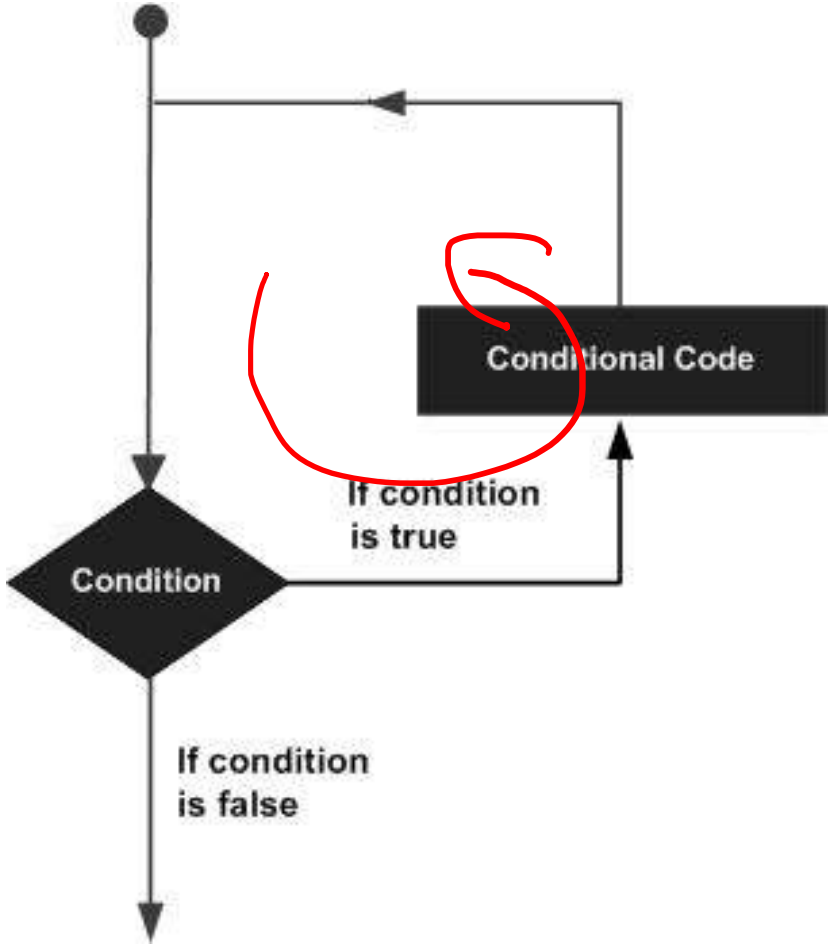
```
Console.WriteLine("Counting Up: " + 0);
Console.WriteLine("Counting Up: " + 1);
Console.WriteLine("Counting Up: " + 2);
Console.WriteLine("Counting Up: " + 3);
Console.WriteLine("Counting Up: " + 4);
```

```
for ( int counter=0;counter<5; counter=counter+1)
{
   Console.WriteLine("Counting Up: "+counter);
}
```

Topics:

- What are Loops?
  - https://www.tutorialspoint.com/csharp/csharp_loops.htm





Topics:

- Rounding:

```
float value = 0.000001234567f;
Math.Round(value, 8);
```

Topics:

- Rounding:

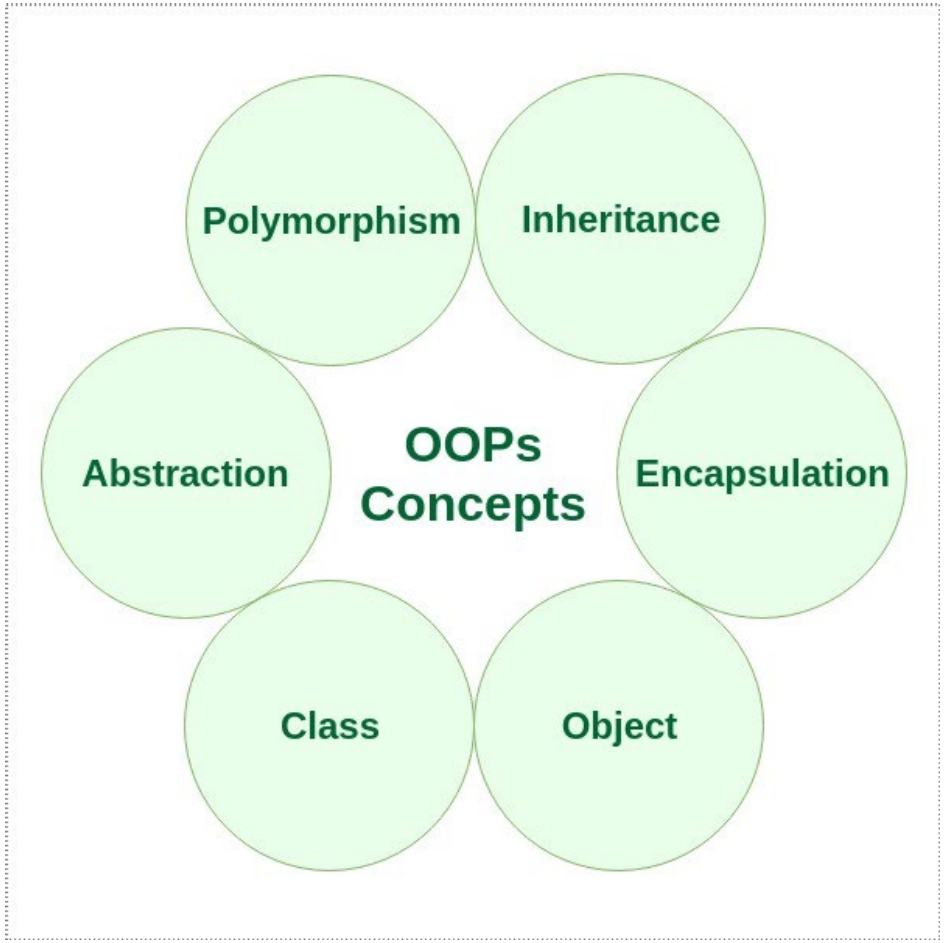
```
float value = 0.000001234567f;
Math.Round(value, 8);
```

```
float value = 0.000001234567f;
value = ((int)(value * 100000000));
value /= 100000000.0f;
```

OOD

Concepts:

- **Polymorphism**: an object can have many forms
- **Inheritance**: an object can inherit traits/features from a parent
- **Encapsulation**: functionality is clustered and hidden from the outside
- **Abstraction**: displaying only essential functionality to the user (**interface**)
- **Object**: basic active structure that programs work with (state, behaviour)
- **Class**: Bluprint for objects, concept for functionality of objects, defines **modifiers**, **name**, **inheritance**, **behaviour**



Discussing a Class in C#

```
using System;
namespace Crawler
{
    public class CMDCrawler
    {
        public char action = 'N';
        private bool active = false;

        public CMDCrawler()
        {}

        public void ProcessUserInput(string input){}
        public int[] GetPlayerPosition(){}
        public int GetPlayerAction(){}

        static void Main(string[] args)
        {
            CMDCrawler crawler = new CMDCrawler();
        }
    }
}
```

Access Modifers

- Public
- Internal
- Protected
- Private

https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/access-modifiers

Inheritance in C#

```
public class Entity
    {
        protected int posX = 0;
        protected int posY = 0;
        protected float health = 0;
        protected bool canMove = false;

        public int[] getPosition() {}
    }

 public class Monster : Entity
    {
        public Monster()
        {
            canMove = true;
        }

        protected int damage = 0;

    }
```

https://www.w3schools.com/cs/cs_inheritance.asp

Topics:

- Rounding and Packages/Namespaces:
- https://www.tutorialspoint.com/Packages-in-Chash
- https://www.tutorialspoint.com/csharp/csharp_namespaces.htm

Libraries vs Executables

https://docs.microsoft.com/en-us/dotnet/core/tutorials/library-with-visual-studio
https://docs.microsoft.com/en-us/dotnet/core/tools/dotnet-add-reference