ECE 314 Computer architecture laboratory Object oriented programming C/C++ **assistive notes**

Slides and material adopted from MIT OPEN COURSEWARE – Introduction to C Memory management and C++ object oriented programming.

Outline of presentation

- Object oriented programming in general
- Classes
- Fields and methods
- Objects
- Constructors
- Representation invariant
- Polymorphism
- Inheritance

Use of objects and levels of abstraction

- Binary code
- Assembly language
- Procedural language
- Object oriented language
- Declarative languages

- Low level languages are closer to hardware and more efficient.
- High level languages are simpler to write and support.

Use of objects and levels of abstraction

- An object is a model of an element
  - Characteristics
  - Behaviors

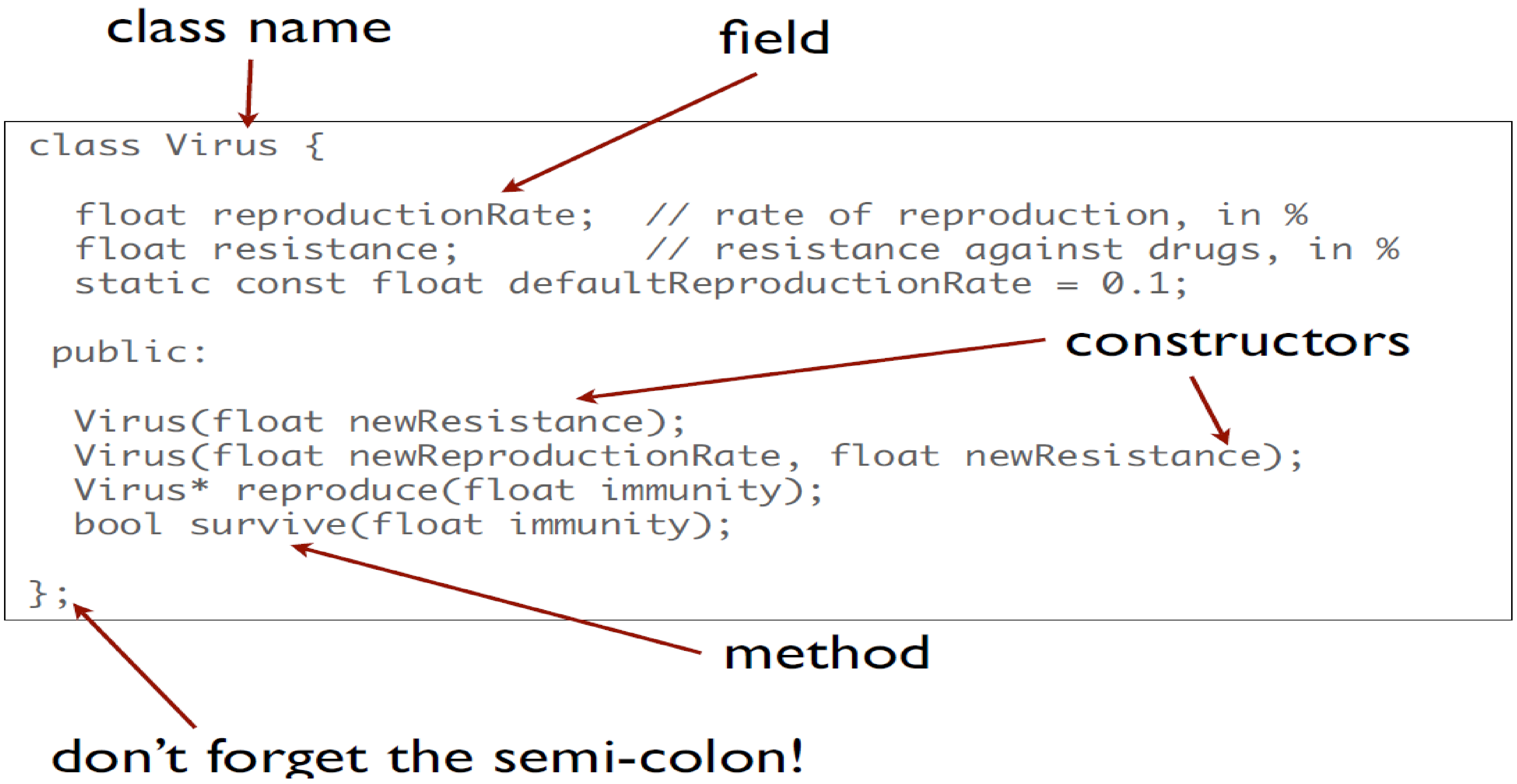
- Example(MIPS 32 Microprocessor)
  - Objects → Register file, ALU, Control unit etc.
  - Characteristics → Number of registers, Bit representation accuracy, number of multiply/accumulate units etc.
  - Responsibilities→ Update register X, execute the addition from command file, override value B in memory location Y etc.

Classes

- Class → Cookie cutter

- Object → cookie

Classes: Declaration



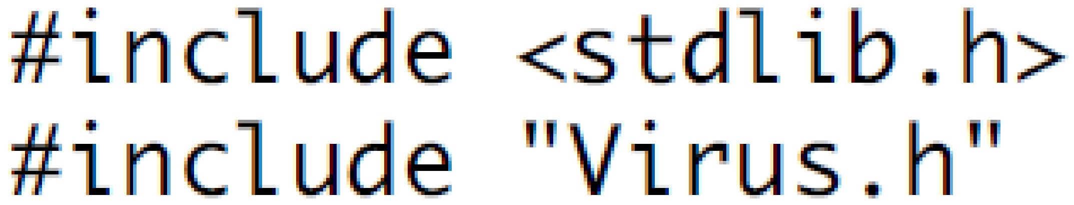
```
class Virus {

   float reproductionRate;  // rate of reproduction, in %
   float resistance;        // resistance against drugs, in %
   static const float defaultReproductionRate = 0.1;

 public:

   Virus(float newResistance);
   Virus(float newReproductionRate, float newResistance);
   Virus* reproduce(float immunity);
   bool survive(float immunity);

};
```

Slide adopted from MIT OPEN COURSEWAVE (Introduction to C memory management and C++object oriented programming

Classes: Declaration

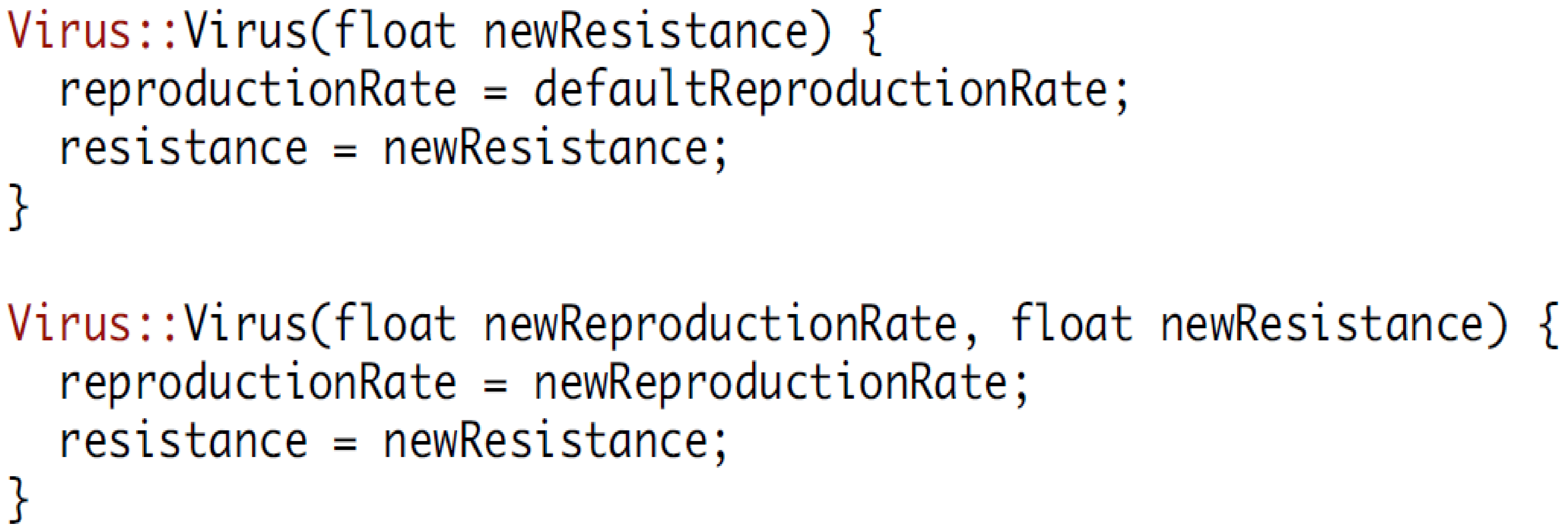
- Fields→ Characteristics
- Methods → Responsibilities
- Constructors → Special syntax, no return type
- Access control of variables:
  - Private: Only accessible inside the class
  - Public: Accessible by anyone

Classes: Header Definition



```
#include <stdlib.h>
#include "Virus.h"
```

Classes: Constructor Definition



```
Virus::Virus(float newResistance) {
  reproductionRate = defaultReproductionRate;
  resistance = newResistance;
}

Virus::Virus(float newReproductionRate, float newResistance) {
  reproductionRate = newReproductionRate;
  resistance = newResistance;
}
```

Slide adopted from MIT OPEN COURSEWAVE (Introduction to C memory management and C++object oriented programming

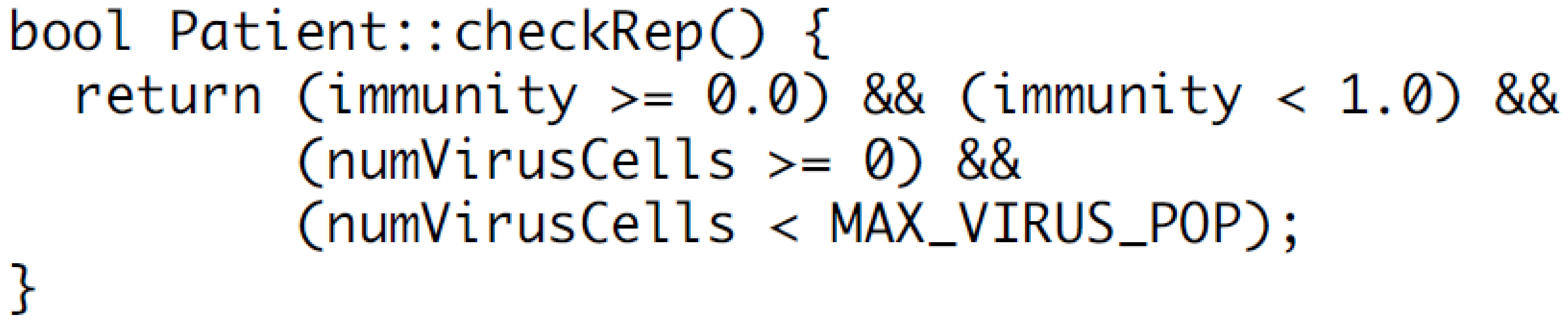
Classes: Method Definition

```
// Returns true if this virus cell survives,
// given the patient's immunity
bool Virus::survive(float immunity) {

  // If the patient's immunity is too strong,
  // then this cell cannot survive
  if (immunity > resistance)
    return false;

  return true;
}
```

Slide adopted from MIT OPEN COURSEWAVE (Introduction to C memory management and C++object oriented programming

Representation invariant

- Statements concerning characteristics of objects
- Defines what makes an object valid
- checkRep checks if rep. invariant is true



```
bool Patient::checkRep() {
  return (immunity >= 0.0) && (immunity < 1.0) &&
         (numVirusCells >= 0) &&
         (numVirusCells < MAX_VIRUS_POP);
}
```

Slide adopted from MIT OPEN COURSEWAVE (Introduction to C memory management and C++object oriented programming

Inheritance

- A class defines a set of objects, a type.
- A subtype inherits characteristics and behaviors of its base type.
- Access control
  - Public: Accessible by anyone.
  - Protected: Accessible inside the class and by all of its subclasses.
  - Private: Accessible only inside the class, not including its subclasses.

Inheritance

```
#include <iostream>
using namespace std;

class GFG {
public:
        void call_Function() // function that call print
        {
                print();
        }
        void print() // the display function
        {
                cout << "Printing the Base class Content" << endl;
        }
};
```

Slide adopted from Geeksforgeeks

Inheritance

```
class GFG2 : public GFG // GFG2 inherit a publicly
{
public:
        void print() // GFG2's display
        {
                cout << "Printing the Derived class Content"
                        << endl;
        }
};
int main()
{
        GFG geeksforgeeks; // Creating GFG's pbject
        geeksforgeeks.call_Function(); // Calling call_Function
        GFG2 geeksforgeeks2; // creating GFG2 object
        geeksforgeeks2.call_Function(); // calling call_Function

        // for GFG2 object
        return 0;
}
```

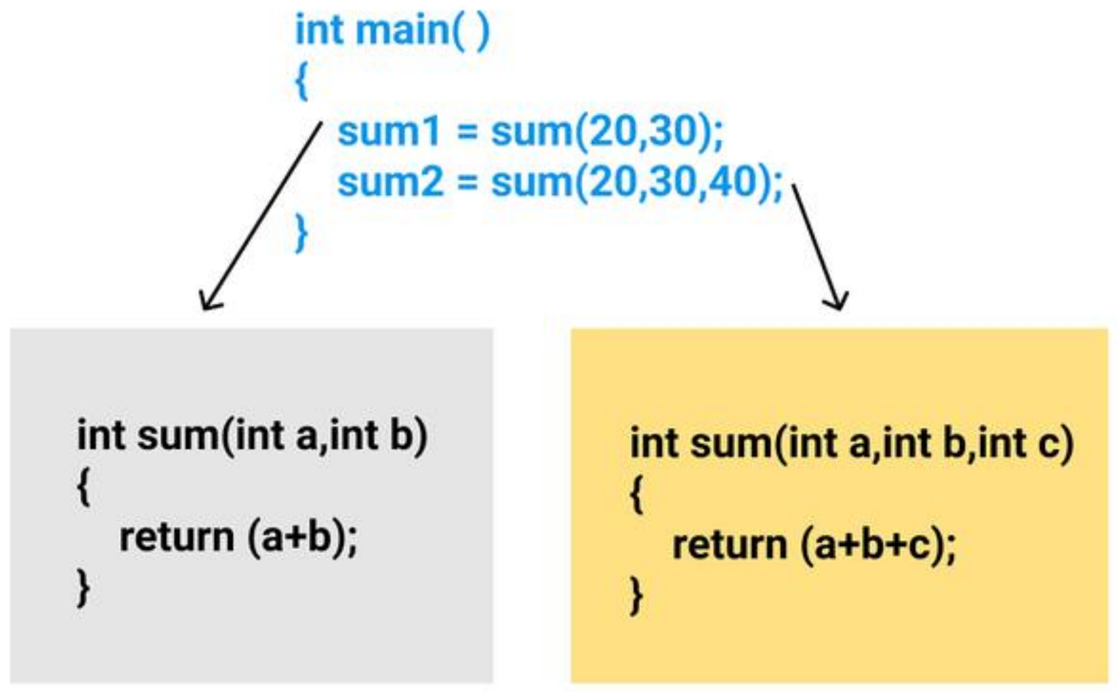
Slide adopted from Geeksforgeeks

Inheritance Output

```
Printing the Base class Content
Printing the Base class Content
```

Polymorphism

- Ability of a type X object to act line another object of type Y



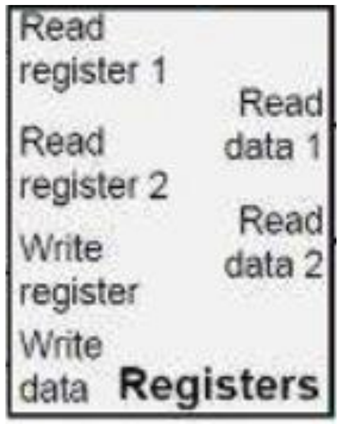
```
int main( )
{
    sum1 = sum(20,30);
    sum2 = sum(20,30,40);
}
```

```
int sum(int a,int b)
{
    return (a+b);
}
```

```
int sum(int a,int b,int c)
{
    return (a+b+c);
}
```

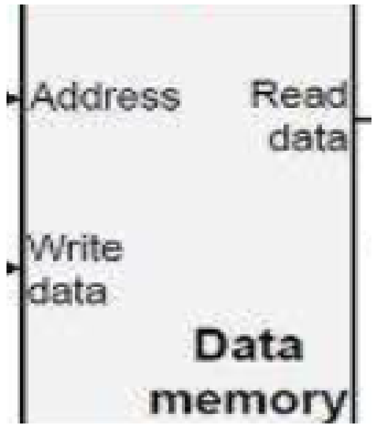
Slide adopted from Geeksforgeeks

OOP Applied on MIPS 32 simulator - Classes



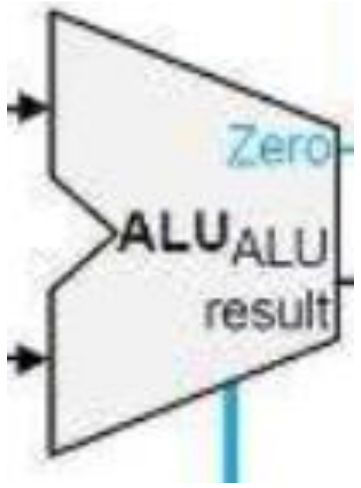
Variables representing all registers

Functions representing all hardware functionality



Variables representing all memory addresses

Functions representing all commands for read and write



Assistive variables for operations

Functions representing all hardware operations e.g. add, subtract etc.

Encapsulated(methods + variables)

Tutorials

https://www.youtube.com/watch?v=wN0x9eZLix4 (1.5 hours tutorial on OOP)

https://www.youtube.com/watch?v=1LGJSRFrxqQ&list=PL43pGnjiVwgTJg7uz8KUGdXRdGKE0W_jN&index=2 (Series of tutorials for OOP)

https://www.youtube.com/watch?v=pTB0EiLXUC8 (4 Basic pillars of OOP – 7 minutes)

https://www.youtube.com/watch?v=8jLOx1hD3_o (full tutorial - 31 hours)

https://www.youtube.com/watch?v=0NwsayeOsd4 (tutorial 30 minutes)

References - Bibliography

- MIT OPEN COURSEWAVE (Introduction to C memory management and C++object oriented programming - https://ocw.mit.edu/courses/6-088-introduction-to-c-memory-management-and-c-object-oriented-programming-january-iap-2010/)
- Thinking in C++ (B. Eckel) (Free pdf edition)
- C++ Programming Language (B. Stroustrup) (Free pdf version via github)
- W3 schools
- GeeksforGeeks