

A job ready bootcamp in C++, DSA and IOT

Function overloading



Saurabh Shukla (MySirG)

Agenda

- ① OOP Principles
- ② Polymorphism
- ③ Function overloading

OOP Key Principle

① Encapsulation

② Data Hiding

③ Abstraction

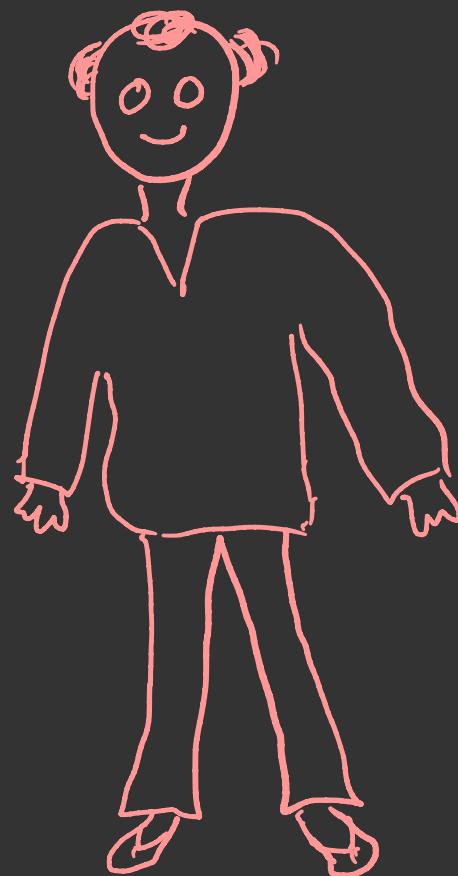
④ Polymorphism

⑤ Inheritance

Polymorphism

Polymorphism is a greek word that means many-shaped.

Real world examples.

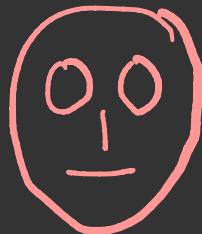


Right

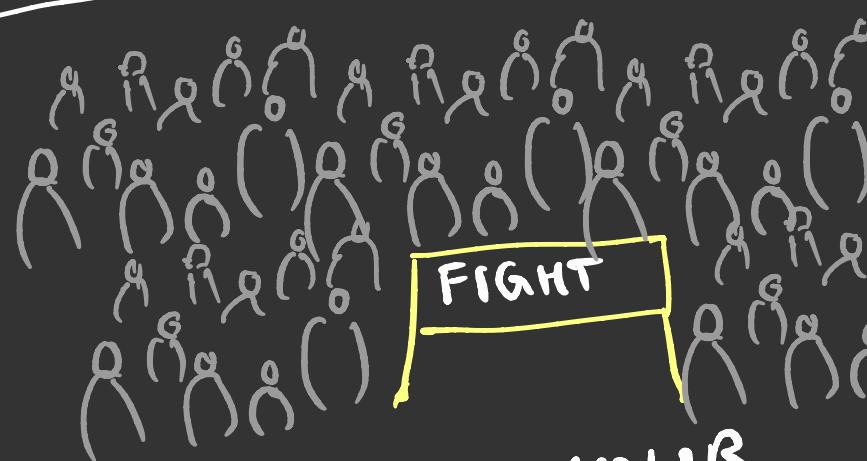


Tree

Take right turn



Am I right?



FIGHT FOR YOUR
RIGHT

How to implement Polymorphism in C++?

- ① Function Overloading
 - ② Operator Overloading
 - ③ virtual function
- RunTime Polymorphism
- Compile Time Polymorphism

Function Polymorphism

Multiple functions sharing same name can be mapped with function call on the basis of arguments at compile time is known as Function Polymorphism or function overloading.

Early Binding

The job of compiler to bind (map) a function call with appropriate function definition is called

Early Binding.

In C language

Function names must be unique

In C++ language

Function signature must be unique

function Signature

① Function name

② Arguments

But not return type

- ① Compiler encounters with a function call.
- ② Compiler searches for the function on the basis of name of the function.
If it finds multiple functions with that name then compiler pick all of them and say them candidates
- ③ In order to select the most appropriate candidate to map with the function call, compiler uses 3-Step rule

Step-1 : Exact match

Step-2 : Type Promotion

Step-3 : Type Conversion

char → int
float → double

①

void f1(int);
float f1(float);

int x=5;
f1(x);

③

void f1(float);
float f1(struct Book);
char x='A';
f1(x);

②

void f1(int);
float f1(float);
char x='A';
f1(x);

④

void f1(float);
void f1(double);
char x='A';
f1(x);

@ error