

C++ in depth

# 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

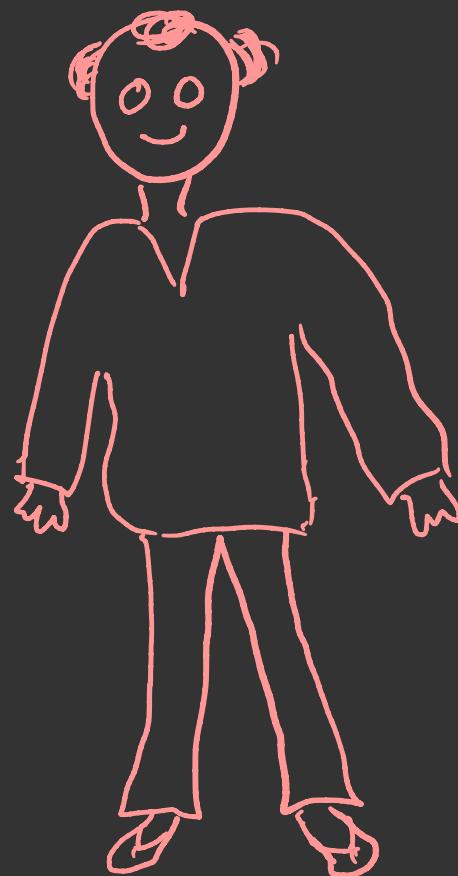
⑤ Inheritance

## Poly morph

## Polymorphism

Polymorphism is a greek word that means many-shaped.

Real world examples.



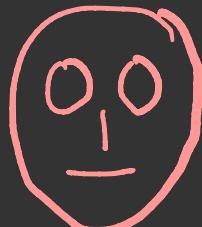
Right



Tree

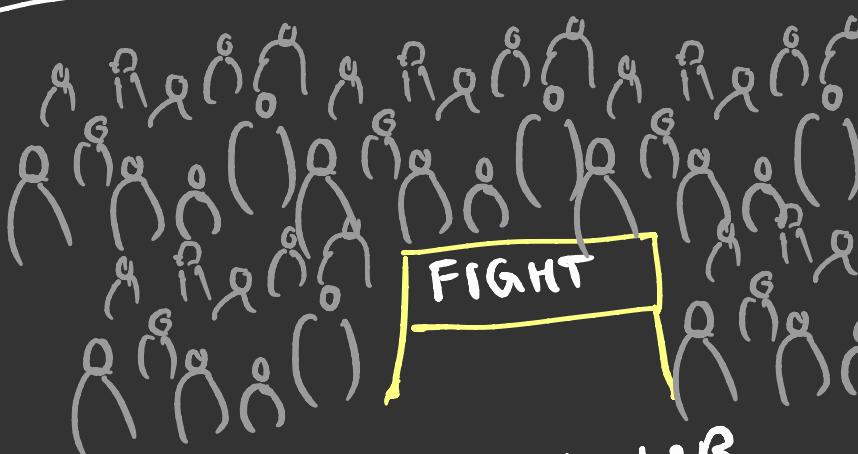
Take right turn

QTR



Am I right?

सही



FIGHT FOR YOUR  
RIGHT

आधिकार

# 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.

## In C language

- Function names must be unique.
- two functions must have different names, even if their prototype is different .

## Function Prototype

① returntype      ② functionName      ③ Argument

```
int f1(float, int);  
void f1 (int);
```

## Early Binding

The job of compiler is 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

Any primitive type → Any primitive type

1

```
void f1(int);  
float f1(float);  
int x=5;  
f1(x);
```

2

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

3

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

4

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

Error

```
int x = 5;  
float y = 6.5;
```

```
cout << x;
```

```
cout << y;
```