C++ in depth

Function overloading

Saurabh Shukla (MySirG)

Agenda

1. OOP principles
2. Polymorphism
3. Function overloading

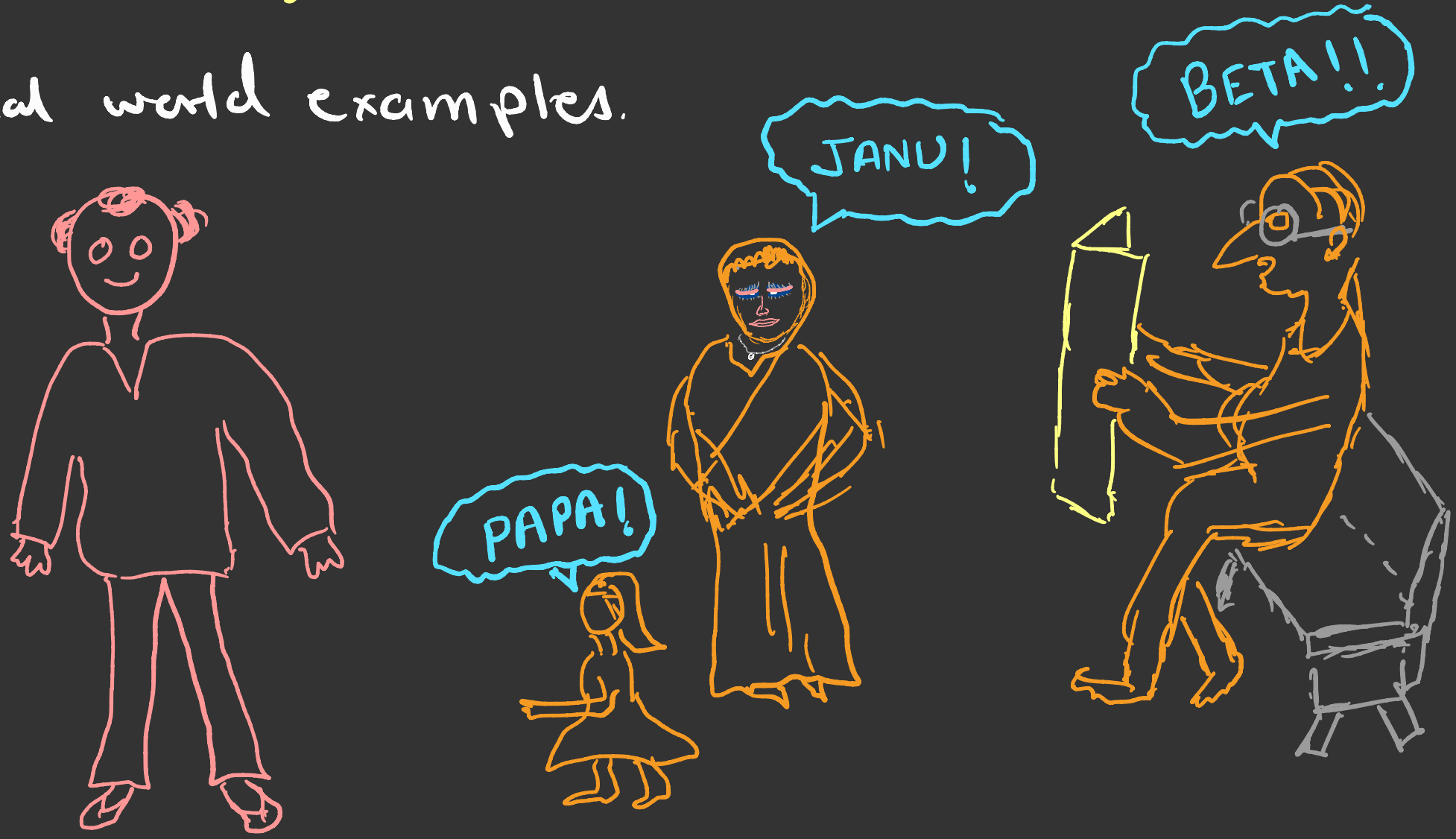
OOP Key Principle

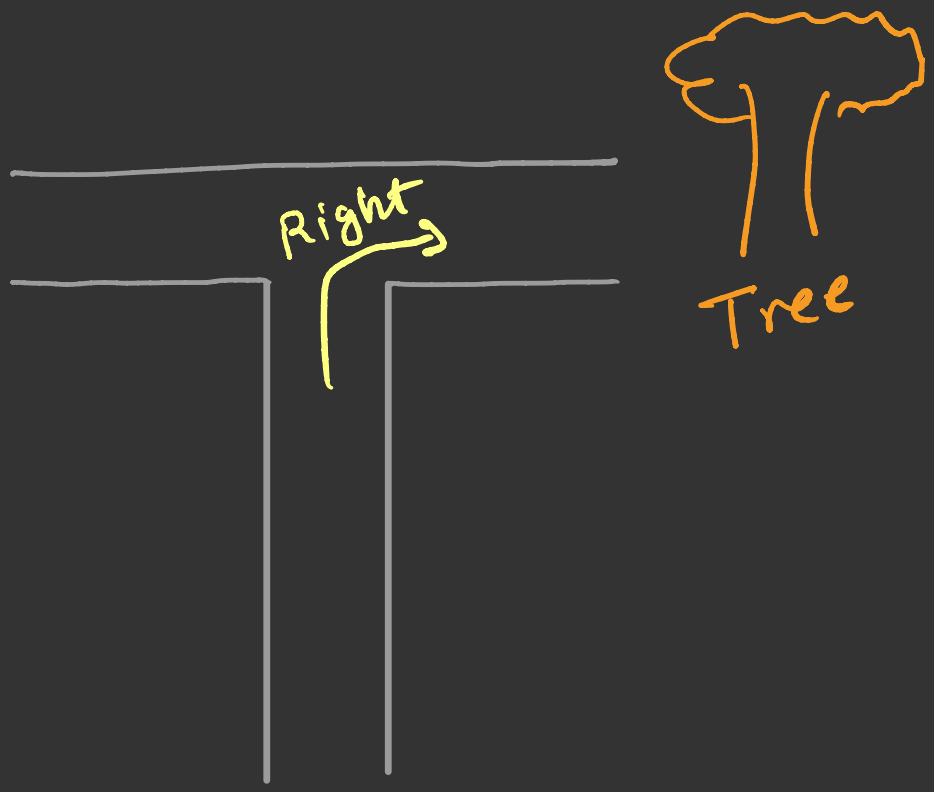
1. Encapsulation
2. Data Hiding
3. Abstraction
4. Polymorphism
5. Inheritance

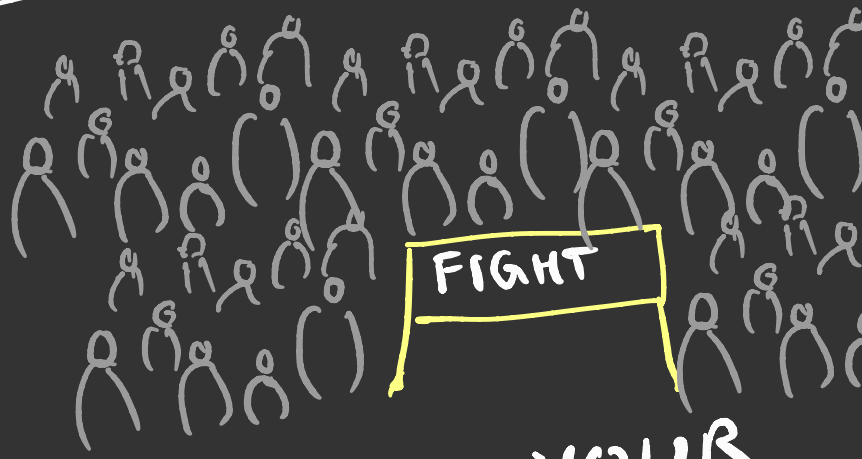
Poly morph Polymorphism

Polymorphism is a greek word that means many-shaped.

Real world examples.







FIGHT FOR YOUR RIGHT

अधिकार

How to implement Polymorphism in C++?

1. Function overloading
2. Operator overloading

} Compile Time Polymorphism

3. virtual function ⟶ RunTime 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 founds 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;
```