Function overloading

Mohammad Tasin

Agenda

- OOP Principles
- Polymorphism
- Function overloading

OOP key Principles

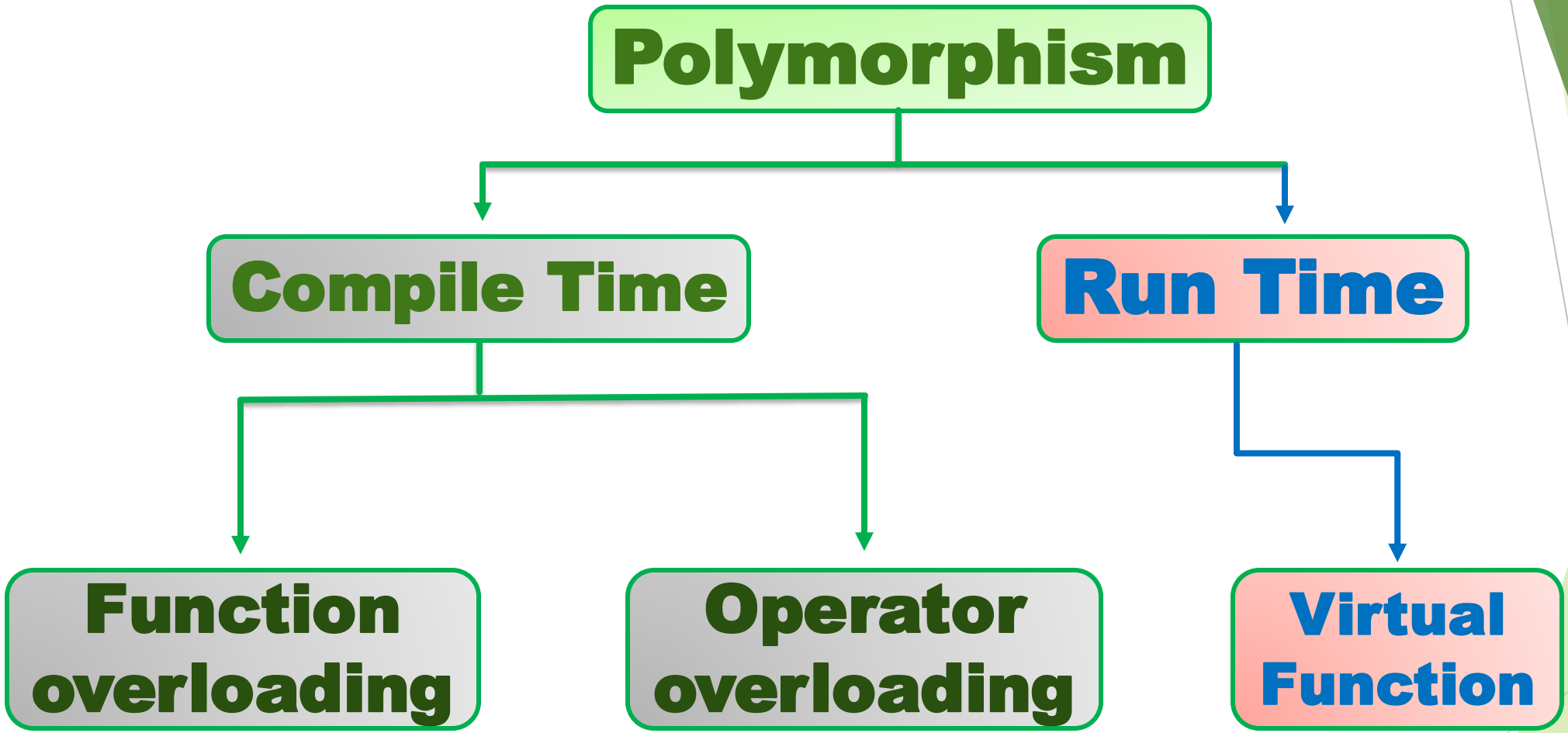
1. Encapsulation
2. Data Hiding
3. Abstraction
4. Polymorphism
5. Inheritance

Polymorphism

- **Polymorphism is a Greek word that means many - Shaped.**

- **Polymorphism** यानी एक शब्द का एक से अधिक मतलब होता हो लेकिन शब्द को जिस जगह पर **use** करते है तो वाह पर उस शब्द का मतलब निश्चित हो

How to implement Polymorphism in C++



Function overloading

- 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

- Compiler encounters with a function call
- 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.
- In order to select the most appropriate candidate to map with the function call, compiler use

  3 - Step Rules

  - Exact match
  - Type Promotion
  - Type Conversion

1. Exact Match

- इसमें **function call** और **function declaration** match होता हो **only** एक **function**

```
void F1(int);
void F1(float);
char F1(char);
int main()
{
    int x = 10;
    F1(x);
}
```

- int → int
- float → float
- double → double
- char → char

2. Type Promotion

- इसमें **type** को प्रोमोट करके **function call** और **function declaration match** होता हो **only** एक **function.**
- इसमें **data loss** नहीं होता है

1. **char → int**
2. **float → double**

```
int F1(int);
void F1(float);
int main()
{
    char x = 'T';
    F1(x);
}
```

3. Type Conversion

- इसमें **data Loss** भी हो सकता हैं

```
void F1(struct Book);
void F1(int a);
int main()
{
    double d1 = 101.999;
    F1(d1);
    return 0;
}
```