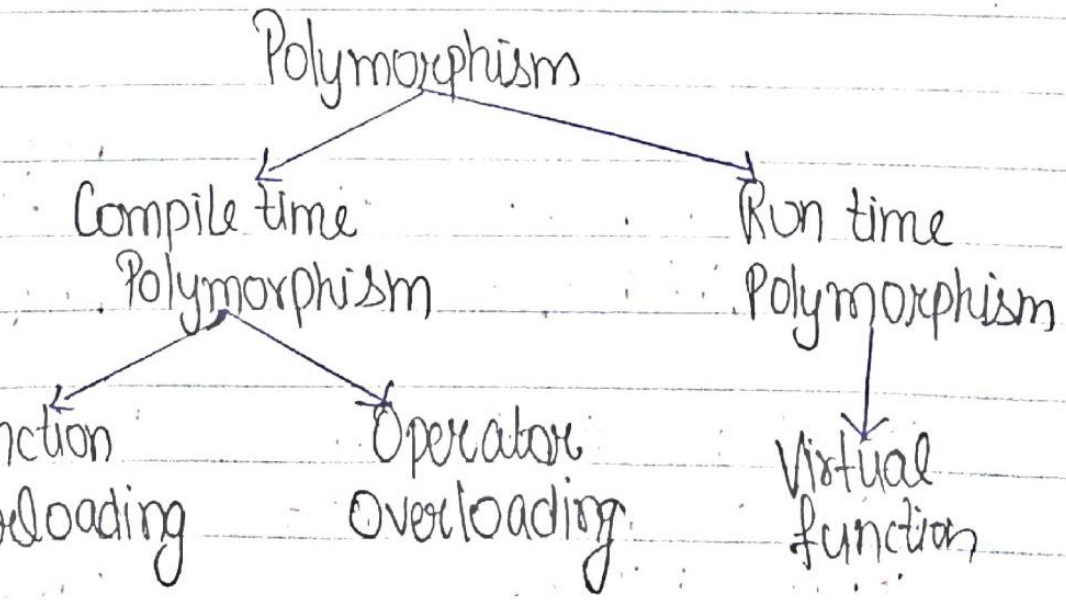
Unit-IV

# Polymorphism

- The term "polymorphism" is the combination of "poly" + "morphism" which means many forms.
- Polymorphism is important OOP Concept.
- Polymorphism have ability to take many forms.
- Real life Example:- A lady behaves like a teacher, in a classroom, mother or daughter in a home and customer in a market. Here, a single person is behaving according to the situations.
- Types of Polymorphism



- Compile Time Polymorphism :- The overloaded function to be invoked is known at the compile time.

→ It is achieved by function overloading and operator overloading which is also known as "static binding or early binding".

- Run time polymorphism :- The function to be invoked is known at run time.

→ It is achieved by method overriding which is also known as "dynamic binding or late binding"

- function Overloading :- function overloading is defined as the process of having two or more function with same name, but different in parameter is known as function overloading in C++.

# Program for function overloading.

```
#include<iostream.h>
#include<conio.h>
class display
{
public:
void show(int i)
{
cout<<"Printing int :"<<i;
}
void show (double f)
{
cout<<"\nPrinting float :"<<f;
}
void show(char c[10])
cout<<"\nPrinting character:"<<c;
}
};

void main()
{
clrscr();
display d;
d.show(4);
```

```
d.show(-2);
d.show("Palvi");
getch();
}
```

Output :- Printing Int : 4
Printing Float : 1.2
Printing Character : Palvi

- ## Operator Overloading :-

Operator overloading is a compile-time polymorphism in which the operator is overloaded to provide the special meaning to user defined data type.

Def: To assign more than one operation on an same operator is known as Operator Overloading.

There are two types of operator overloading :-

→ Unary Operator Overloading
→ Binary Operator Overloading