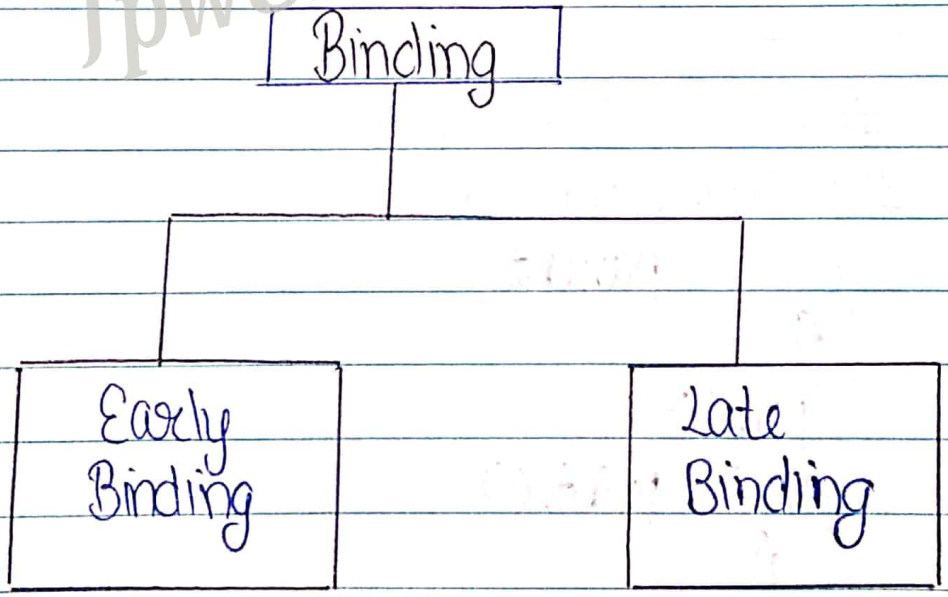
Early Binding and Late Binding

- Binding means linking. for each variables and functions this binding is done.
- The binding is done either compile time or at run time.
- Binding refers to the process of converting identifiers into addresses.



- Binding has two types

1. Early Binding
2. Late Binding

① Early Binding :-

- This is compile time Polymorphism.
- This is also called static binding or compile-time binding.
- In this, by default, c++ matches a function call with correct function definition at compile time.
- Static binding / early binding is achieved using function overloading and operator overloading.

WAP of early Binding

```
#include<iostream.h>
#include<conio.h>
  class parent
   {
    public:
    void print()
     {
    cout<< "Display parent class ";
     }
};
   class child : public parent
```

```
{
public:
void print()
{
cout << "display child class";
}
};

void main()
{
parent* p = new child;
p -> print();
getch();

}
```

Output :- Display Parent class.

2 Late Binding :-

- This is run time polymorphism.
- This is also called dynamic binding or run-time binding.
- If C++ matches a function call with the

correct function definition at run time, is called Late binding.
This is achieved using virtual function.

Program for Late Binding

```
#include<iostream.h>
#include<conio.h>
Class parent
{
public:
virtual void print()
{
cout<<"Display parent class";
}
};

Class child : public parent
{
public:
void print()
{
cout<<"display child class";
}
};

void main()
{
```

```
clrscr();
Parent *p = new child ;
p->print();
getch();
}
```

Output :- display child class.

* Difference between Early Binding and Late Binding.

	Early Binding	Late Binding
1	Early Binding happens at the compile time.	Late Binding happens at the run time.
2.	Its type is static.	Its type in dynamic.
3.	It use function overloading and operator overloading.	Use virtual function.
4.	This is called as compile time polymorphism.	This is called as run time polymorphism.