```
class Test
{
    public static void main(String[] args)
    {
        Calculator c = new Calculator();
        c.add(10,20);
    }
}
```

int,int → Compiler

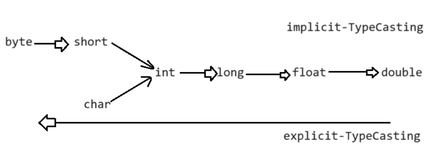
add(int,int)
add(float,float)
add(double,double)

```
class Calculator
{
    public void add(int a,int b){
        System.out.println("int-int argument");
    }
    public void add(float a,float b){
        System.out.println("float-float argument");
    }
    public void add(double a,double b){
        System.out.println("double-double argument");
    }
}
```

"Polymorphism" (False Polymorphism)
1 person ====> Mulitple jobs

add(int,int)
add(float,float)
add(double,double)

Compiler is binding so the polymorphism is
a. Early Binding
b. Static Binding
c. Eager Binding



Overloading

Compiler binding the call based on argument
a. if exact match is found bind the call
b. if exact match not found, perform
implicit typecasting till it reaches to bind
c. upon implicit typecasting still if the call
can't be bind, it would result in "CE"

```
class Test
{
    public static void main(String[] args)
    {
        Calculator c = new Calculator();
        c.add(10,20);
    }
}
```

int,int ------> exactmatch ✗

int,float
float,int
float,float
⇨ Type promotion

```
class Calculator
{
    public void add(int a,float b){
        System.out.println("int-float argument");
    }
    public void add(float a,int b){
        System.out.println("float-int argument");
    }
}
```