```
class Calculator
    class Test
                                                         public static void main(String[] args)
                                                         public void add(float a,float b){
            Calculator c = new Calculator();
                                                             System.out.println("float-float argument");
           c.add(10,20);
                                   Compiler
                  int,int —
                                                          public void add(double a,double b){
                                                             System.out.println("double-double argument");
                                 add(int,int)
                                 add(float,float)
add(double,double) }
    "Polymorphism" (False Polymorphism)
                                                  Compiler is binding so the polymorphism is
       1 person ====> Mulitple jobs
                                                   a. Early Binding
                                                   b. Static Binding
                       add(int,int)
                                                   c. Eager Binding
                       add(float,float)
                        add(double,double)
                                        implicit-TypeCasting
                                                                 Overloading
 byte—∟> short ✓
                                                                    Compiler binding the call based on argument a. if exact match is found bind the call
                        nt ——

characteristic double
                                                                       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"
                                       explicit-TypeCasting
                                                     class Calculator
 class Test
                                                        public void add(int a,float b){
    public static void main(String[] args)
                                                           System.out.println("int-float argument");
       Calculator c = new Calculator();
                                                        public void add(float a,int b){
       c.add(10,20);
                                                           System.out.println("float-int argument");
             int,int -----> exactmatch ⊀
            int,float
float,int 
Type promotion
             float, float
                                          Object(C)
                                                                                 Runnable (I)
                                                                                  Thread
                                                             Character Boolean
              StringBuilder
                               StringBuffer
String
                                                                                         Compiler --->Mother
                                     Byte Short Integer Long Float Double
                                                                                                Father (Object)
                                                                                                      Child
                                                                                          Child
                                                                                                     (StringBuffer)
                                                                                       (String)
```