

```

class BI {
    public void f1() { ... } //A
}

class B2 {
    public void f2(B1 b) { ... } //C
    public void f2(D1 d) { ... } //D
}

class D1 extends BI {
    public void f1() { ... } //B
}

class D2 extends B2 {
    public void f2(B1 b) { ... } //E
    public void f2(D1 d) { ... } //F
}

f3(B2 b) {
    B1 b1 = new B1...
    b.f2(b1); //G
}

f3(D2 d) {
    B1 b1 = new D1...
    d.f2(b1); //H
}

main... {
    B2 b = new B2...
    f3(b); //load G, oride C

    B2 bd = new D2...
    f3(bd); //load G, oride E

    D2 d = new D2
    D2 d = new D2
    f3(d); //load H, oride E
}

```

JAVA

Overloading = Compiler matching class

different argument types

Class Person

public void f1()

class Student extends Person

public void f1()

f2(Person p)

p.f1()

f2(Student s)

s.f1();

// overriding of run-time

Person * p1;

Person * p2;

Student * s1;

p1 = new Person;

p2 = new Student;

s1 = new Student;

Person

// C-A

// C-B

// C-B