1. Is the method properly overloaded?

```java
class X
{
    int method(int i, int d)
    {
        return i+d;
    }

    static int method(int i, double d)
    {
        return (int)(i+d);
    }

    double method(double i, int d)
    {
        return i+d;
    }

    static double method(double i, double d)
    {
        return i+d;
    }
}
```

2. What is the output or outcome?

A)

```java
class X
{
    void method(int a)
    {
        System.out.println("ONE");
    }

    void method(double d)
    {
        System.out.println("TWO");
    }
}

class Y extends X
{
    @Override
    void method(double d)
    {
        System.out.println("THREE");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        new Y().method(100);
    }
}
```

B)

```java
class X
{
    void calculate(int a, int b)
    {
        System.out.println("Class X");
    }
}

class Y extends X
{
    @Override
    void calculate(int a, int b)
    {
        System.out.println("Class Y");
    }
}

class Z extends Y
{
    @Override
    void calculate(int a, int b)
    {
        System.out.println("Class Z");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        X x = new Y();

        x.calculate(10, 20);

        Y y = (Y) x;

        y.calculate(50, 100);

        Z z = (Z) y;

        z.calculate(100, 200);
    }
}
```

C)

```
class A
{
    public A(int i)
    {
        System.out.println(1);
    }

    public A()
    {
        this(10);

        System.out.println(2);
    }

    void A()
    {
        A(10);

        System.out.println(3);
    }

    void A(int i)
    {
        System.out.println(4);
    }
}


public class MainClass
{
    public static void main(String[] args)
    {
        new A().A();
    }
}
```

D)

```
class A
{
    void myMethod(Object o, Double D)
    {
        System.out.println(1);
    }

    void myMethod(Integer I, Number N)
    {
        System.out.println(2);
    }
}

class B extends A
{
    void myMethod(Float F, Double D)
    {
        System.out.println(3);
    }

    void myMethod(Double D, Integer I)
    {
        System.out.println(4);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        B b = new B();

        b.myMethod(11.11, 0000);

        b.myMethod(8778, 3223);

        b.myMethod(2.3*1.2, 4.1*1.4);

        b.myMethod((float)23.56, 21.45);
    }
}
```

E)

```
class ABC
{
    void methodABC()
    {
        new XYZ().methodXYZ();
    }
}

class XYZ extends ABC
{
    void methodXYZ()
    {
        methodABC();
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        new ABC().methodABC();
    }
}
```

F)

```
class ABC
{
    void methodABC()
    {
        System.out.println(111);
    }

    void methodABC(int i)
    {
        System.out.println(222);
    }
}

class XYZ extends ABC
{
    @Override
    void methodABC(int i)
    {
        System.out.println(333);
    }

    @Override
    void methodABC()
    {
        System.out.println(444);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        ABC abc = new XYZ();

        abc.methodABC(10);

        abc.methodABC();
    }
}
```