

# Classes and Object

1

```
class R {  
    R() {  
        System.out.print("R");  
    }  
    R(int x) {  
        System.out.print("S");  
    }  
}  
class S extends R {  
    S() {  
        super(10);  
        System.out.print("T");  
    }  
}  
class Test {  
    public static void main(String[] args) {  
        S obj = new S();  
    }  
}
```

- ☐ Compilation error
- ☐ S
- ☐ RST

☒ ST

Predict the output -

```
class ConstructorChaining {
    ConstructorChaining(){
        this(20);
        System.out.println("Default");
    }
    ConstructorChaining(int x) {
        System.out.println("1 parameter");
        System.out.println("x = " + x);
    }
    ConstructorChaining(int x, int y) {
        this();
        System.out.println("2 parameters");
        System.out.println("x = " + x + " and y =" + y +
". "
        + "Sum = " + (x + y));
    }
    public static void main(String... a) {
        ConstructorChaining obj = new
ConstructorChaining(11,12);
    }
}
```

- ☐ 2 parameters x = 11 and y =12. Sum = 23  
Default 1 parameter x = 20
- ☐ Default 1 parameter x = 20 2 parameters x = 11 and y =12. Sum = 23
- ☐ 2 parameters x = 11 and y =12. Sum = 23

- ☒ 1 parameter x = 20 Default 2 parameters x = 11 and y = 12. Sum = 23

Which of these keywords is used to refer to member of base class from a sub class?

☒ this

☐ None of the mentioned

☐ upper`

☒ super

**Incorrect**

Predict the output -

```
class Parent{
    Parent(){
        System.out.println("Default parent");
    }
}
class Child extends Parent{
    Child(){
        System.out.println("Default child");
    }
}
class Main{
    public static void main(String[] args) {
        Child child = new Parent();
    }
}
```

- ☐ Default child
- ☐ Default parent
- ☒ code does not compile
- ☐ None

**Correct**

What is the process of defining a method in a subclass having same name & type signature as a method in its superclass?

- ☐ Method overloading
- ☐ None of the mentioned
- ☒ Method overriding
- ☐ Method hiding

**Correct**