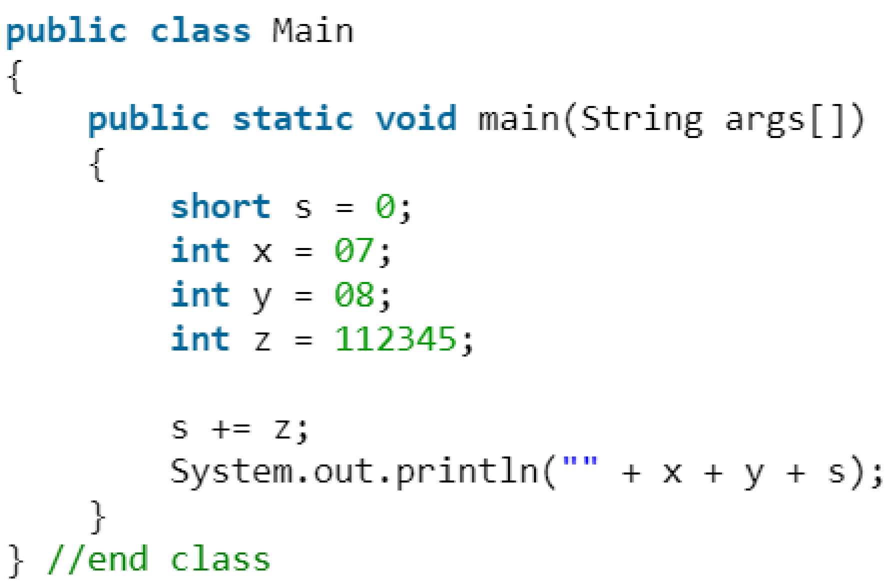
Snippet 2

Total points 7/24 ?

✗ ···/1



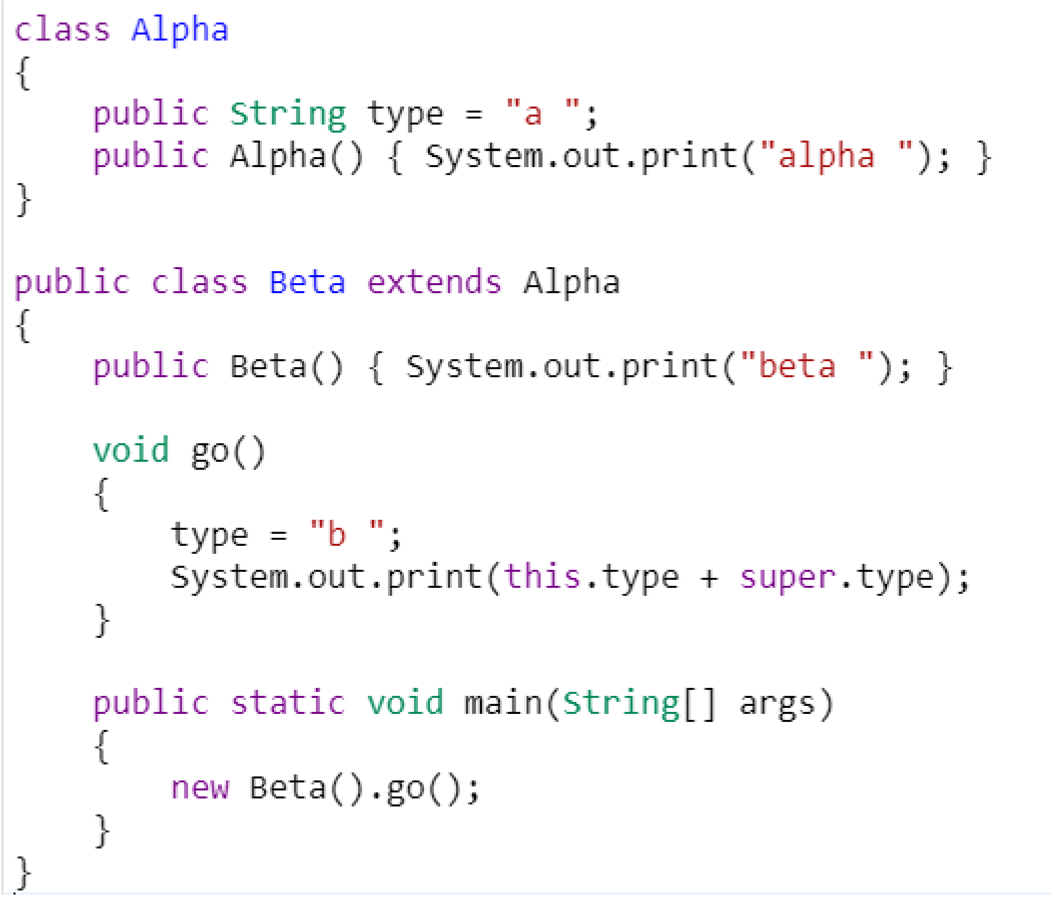
```
public class Main
{
    public static void main(String args[])
    {
        short s = 0;
        int x = 07;
        int y = 08;
        int z = 112345;

        s += z;
        System.out.println("" + x + y + s);
    }
} //end class
```

Error ✗

Correct answer

Line 8 Compiler Error

✓ 1/1



```
class Alpha
{
    public String type = "a ";
    public Alpha() { System.out.print("alpha "); }
}

public class Beta extends Alpha
{
    public Beta() { System.out.print("beta "); }

    void go()
    {
        type = "b ";
        System.out.print(this.type + super.type);
    }

    public static void main(String[] args)
    {
        new Beta().go();
    }
}
```

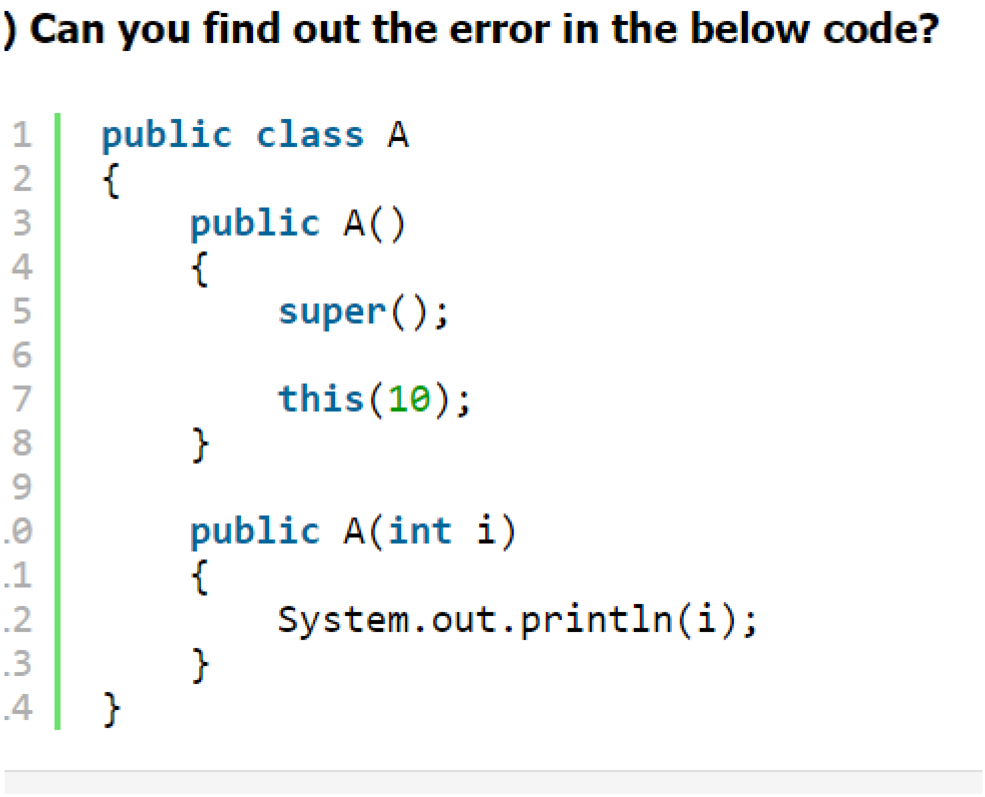
● A) alpha beta b b ✓

○ B) alpha beta a b

○ C) beta alpha b b

○ D) beta alpha a b

✕ ···/1



) Can you find out the error in the below code?

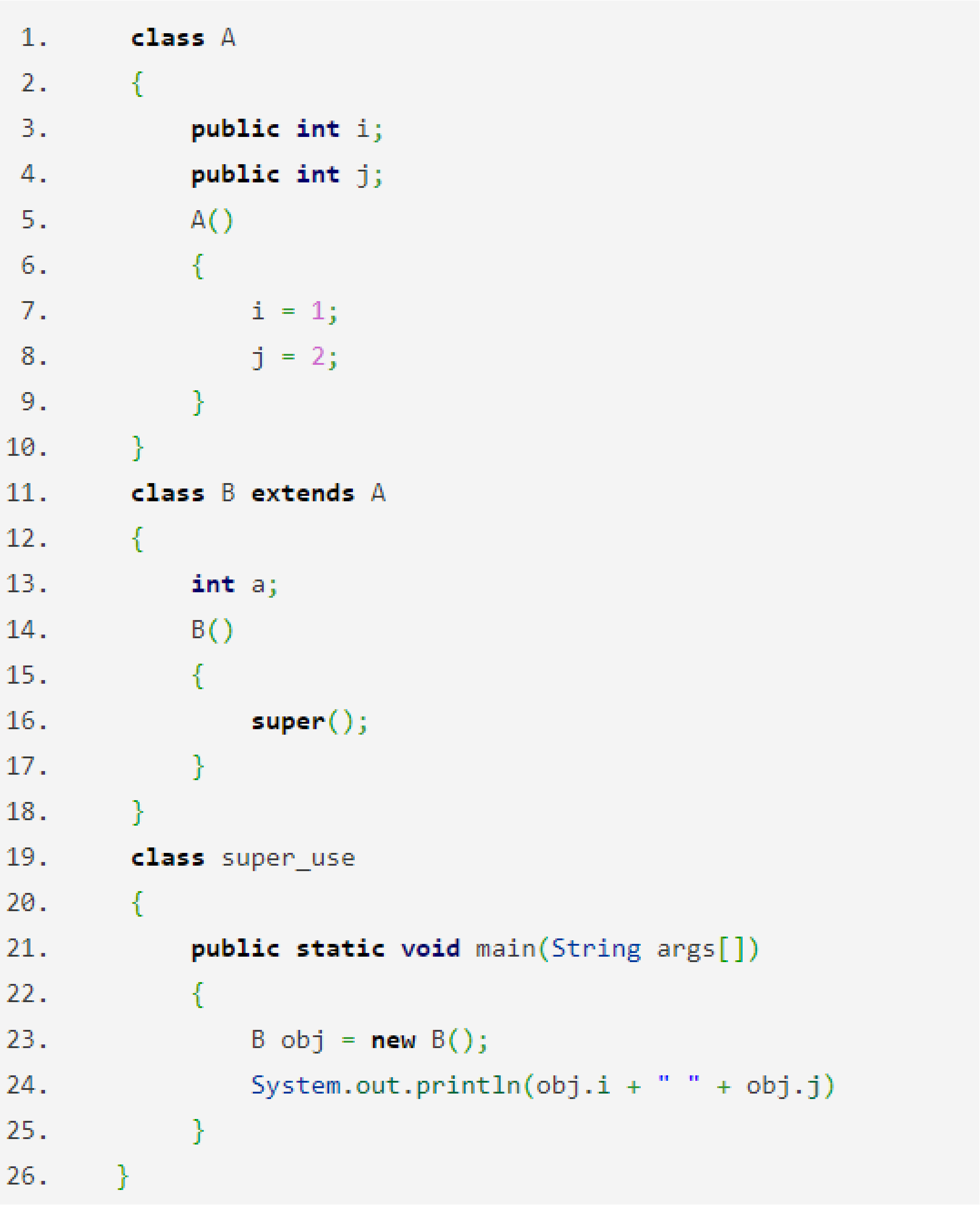
```java
public class A
{
    public A()
    {
        super();

        this(10);
    }

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

this is below

Feedback

A constructor can have either super() or this() but not both.

✓ 1/1



```
class A
{
    public int i;
    public int j;
    A()
    {
        i = 1;
        j = 2;
    }
}
class B extends A
{
    int a;
    B()
    {
        super();
    }
}
class super_use
{
    public static void main(String args[])
    {
        B obj = new B();
        System.out.println(obj.i + " " + obj.j)
    }
}
```

● a) 1 2 ✓

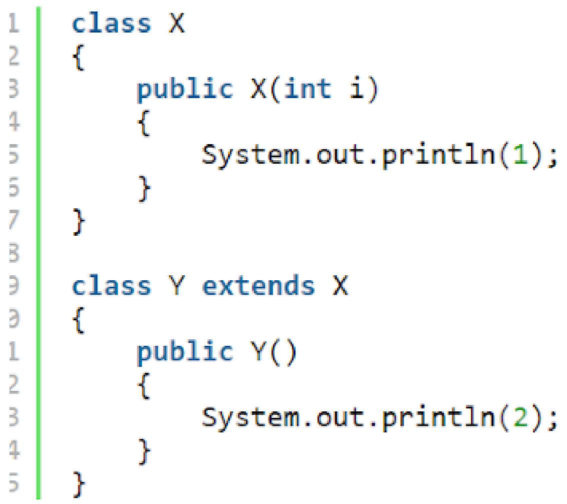
○ b) 2 1

○ c) Runtime Error

○ d) Compilation Error

✗ …/1

Below code is showing compile time error. Can you suggest the corrections?



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

class Y extends X
{
    public Y()
    {
        System.out.println(2);
    }
}
```

no main function ✗

✗ You know that compiler will keep super() calling statement implicitly as a first statement in every constructor. What happens if we write this() as a first statement in our constructor? …/1

It will run ✗

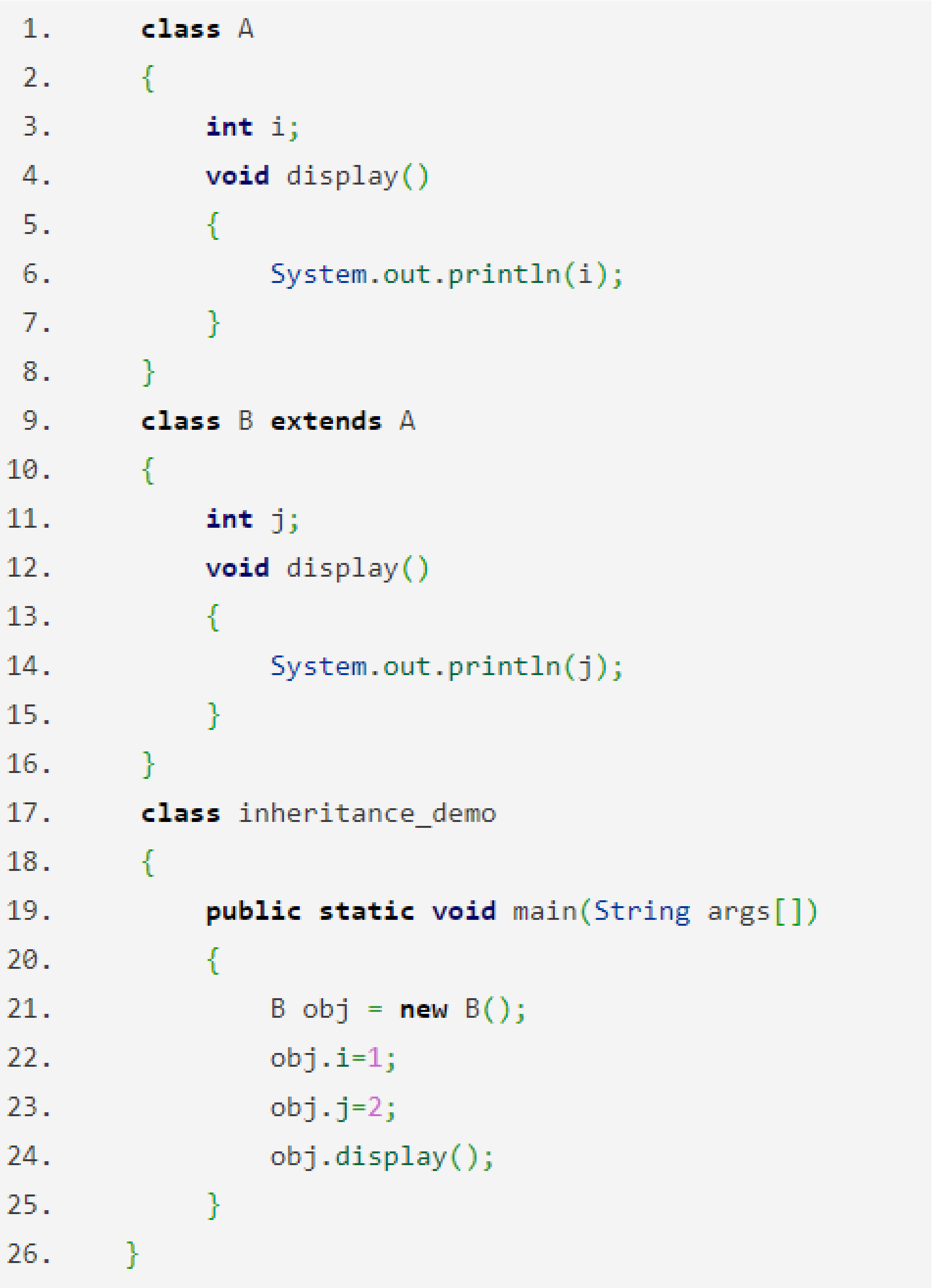
Correct answer

Correct answer

Feedback

Compiler will not keep super() if you are writing this() as a first statement in your constructor.

✓ 1/1



```
class A
{
    int i;
    void display()
    {
        System.out.println(i);
    }
}
class B extends A
{
    int j;
    void display()
    {
        System.out.println(j);
    }
}
class inheritance_demo
{
    public static void main(String args[])
    {
        B obj = new B();
        obj.i=1;
        obj.j=2;
        obj.display();
    }
}
```

○ a) 0

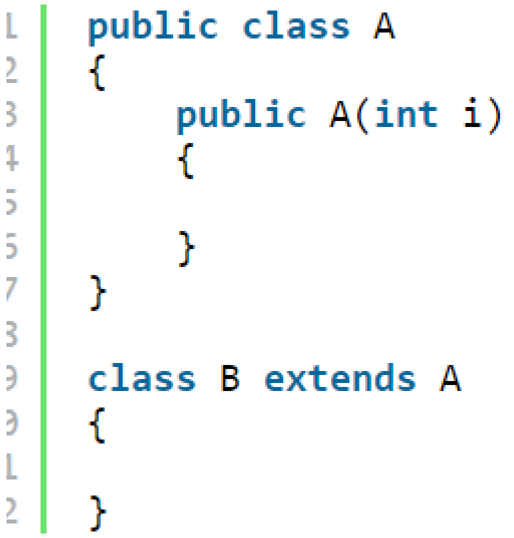
○ b) 1

● c) 2 ✓

○ d) Compilation Error

…/1

Can you find out the error in the below code?



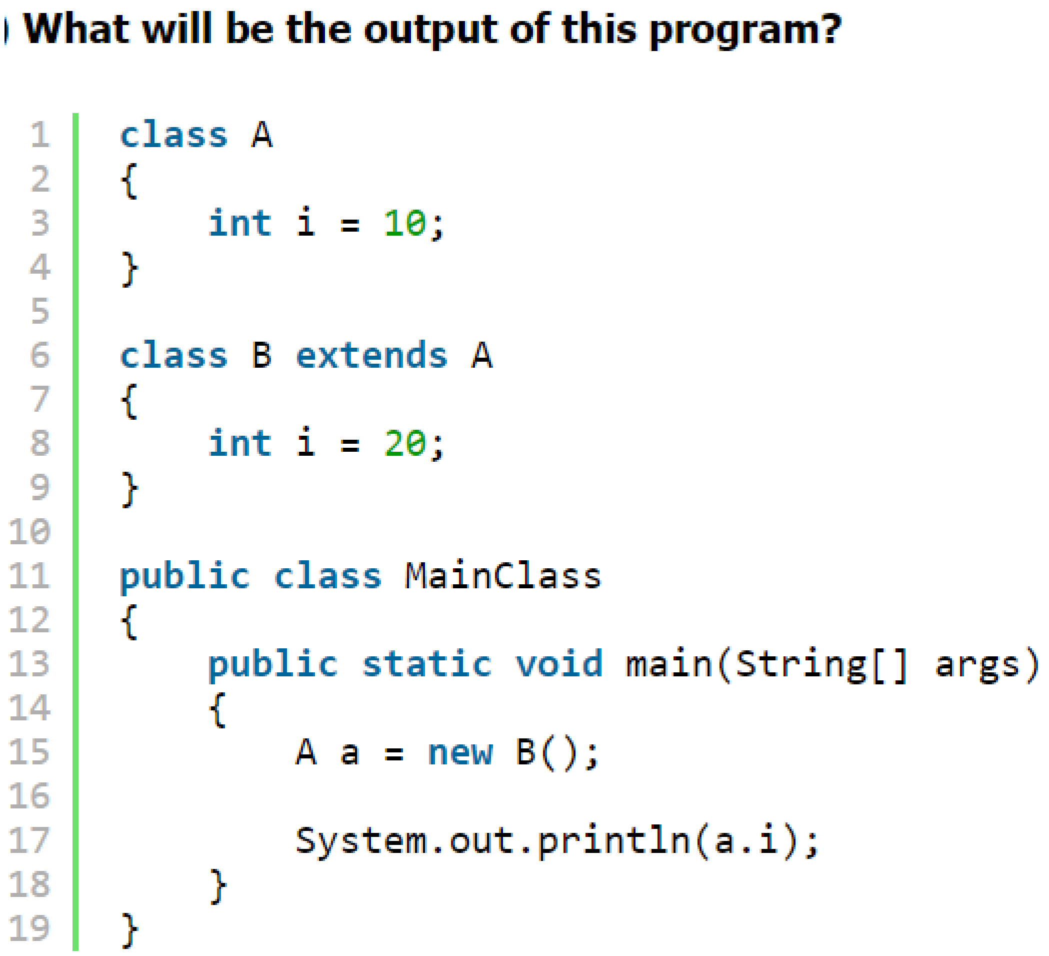
```
public class A
{
    public A(int i)
    {

    }
}

class B extends A
{

}
```

no main method

Feedback

Class B doesn't have any constructors written explicitly. So, compiler will keep default constructor. In that default constructor, there will be a calling statement to super class constructor (super()). But, it is undefined in Class A.To remove the errors, either define A() constructor in class A or write B() constructor in class B and call super class constructor explicitly.

✕ …/1



What will be the output of this program?

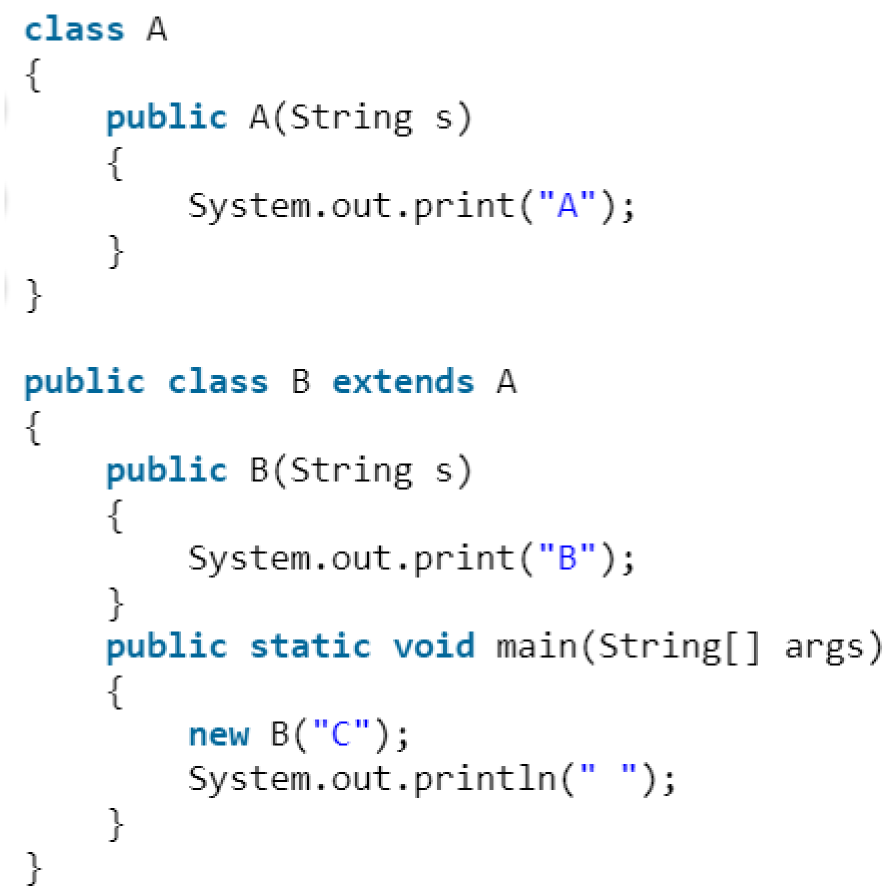
```java
class A
{
    int i = 10;
}

class B extends A
{
    int i = 20;
}

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

        System.out.println(a.i);
    }
}
```

error ✕

Correct answer

10

✗ …/1



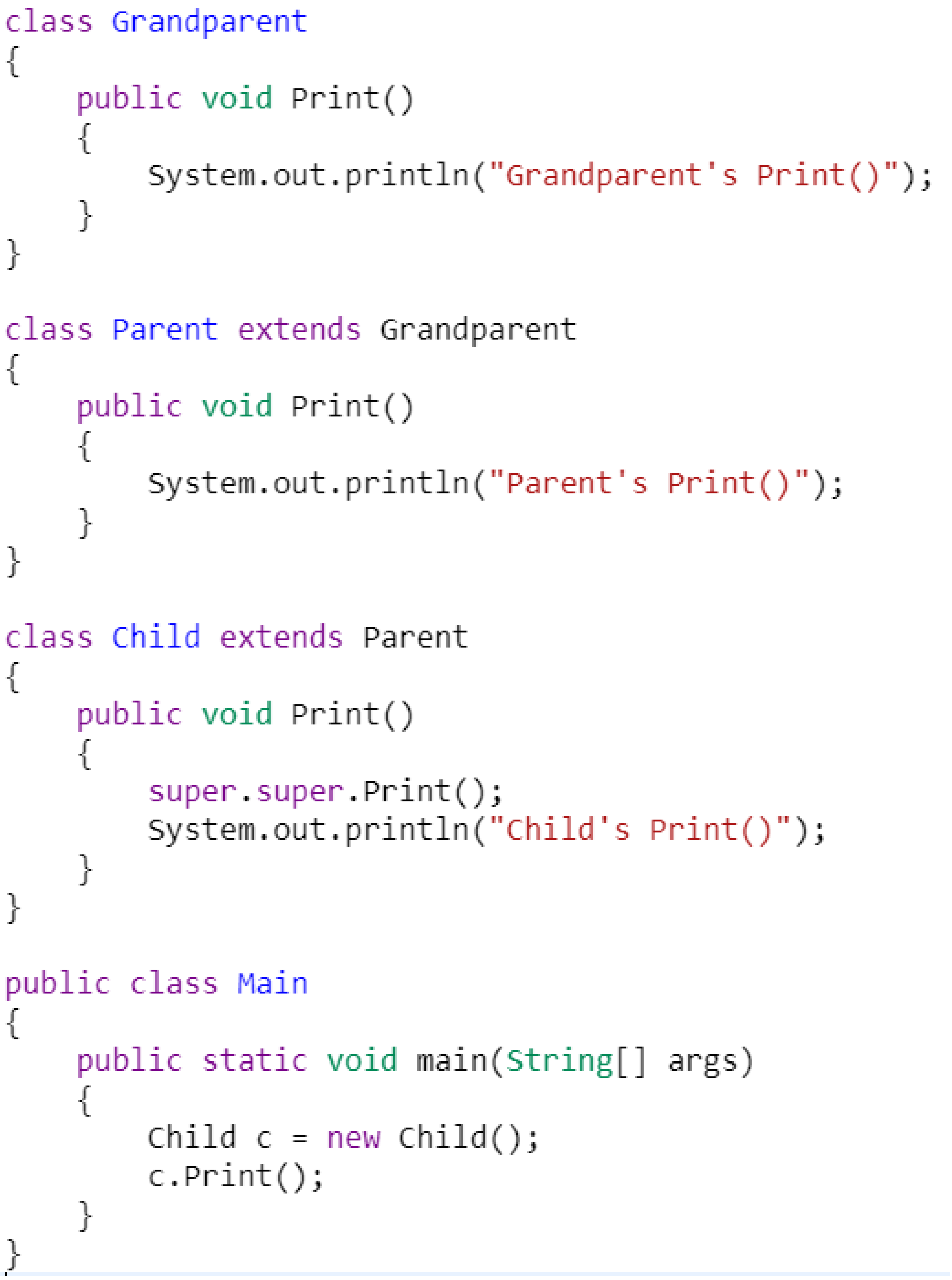
```
class A
{
    public A(String s)
    {
        System.out.print("A");
    }
}

public class B extends A
{
    public B(String s)
    {
        System.out.print("B");
    }
    public static void main(String[] args)
    {
        new B("C");
        System.out.println(" ");
    }
}
```

AB ✗

Correct answer

Compilation Error

✕ …/1



```
class Grandparent
{
    public void Print()
    {
        System.out.println("Grandparent's Print()");
    }
}

class Parent extends Grandparent
{
    public void Print()
    {
        System.out.println("Parent's Print()");
    }
}

class Child extends Parent
{
    public void Print()
    {
        super.super.Print();
        System.out.println("Child's Print()");
    }
}

public class Main
{
    public static void main(String[] args)
    {
        Child c = new Child();
        c.Print();
    }
}
```

Grandparent's Print() Child's Print() ✕

Correct answer

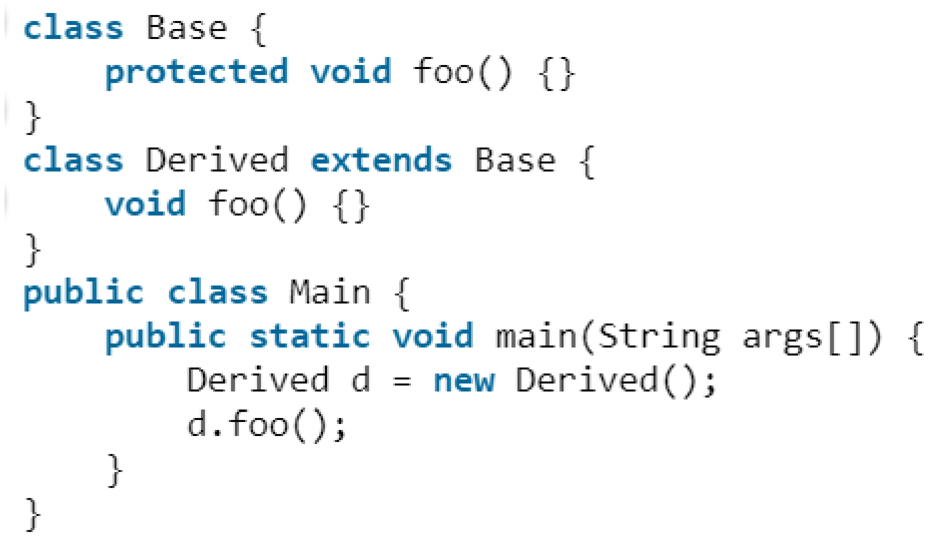
Compilation Error

✓ Private members of a class are inherited to sub class. True or false? 1/1

○ True

● False ✓

✗ * ···/1



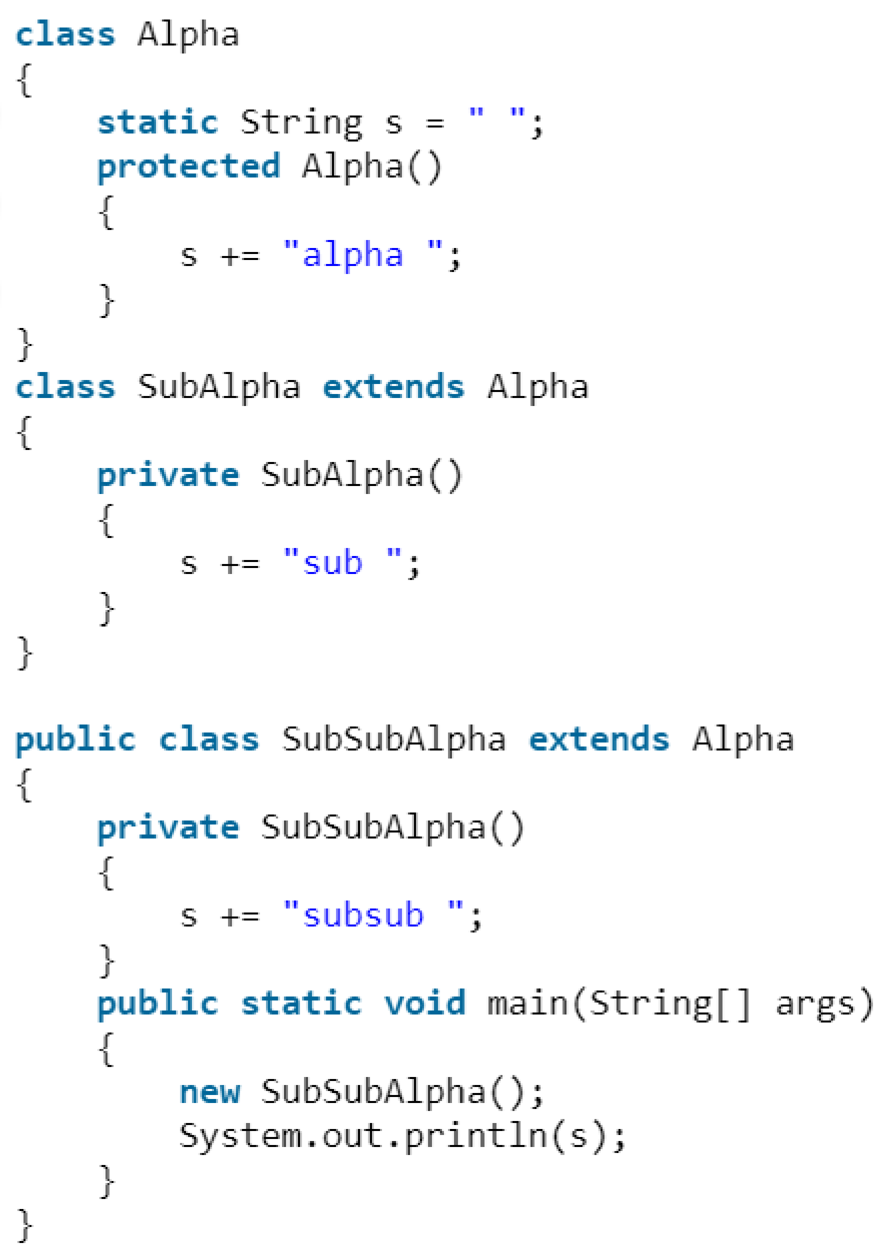
```
class Base {
    protected void foo() {}
}
class Derived extends Base {
    void foo() {}
}
public class Main {
    public static void main(String args[]) {
        Derived d = new Derived();
        d.foo();
    }
}
```

Derived foo ✗

Correct answer

Compiler Error

✓ 1/1



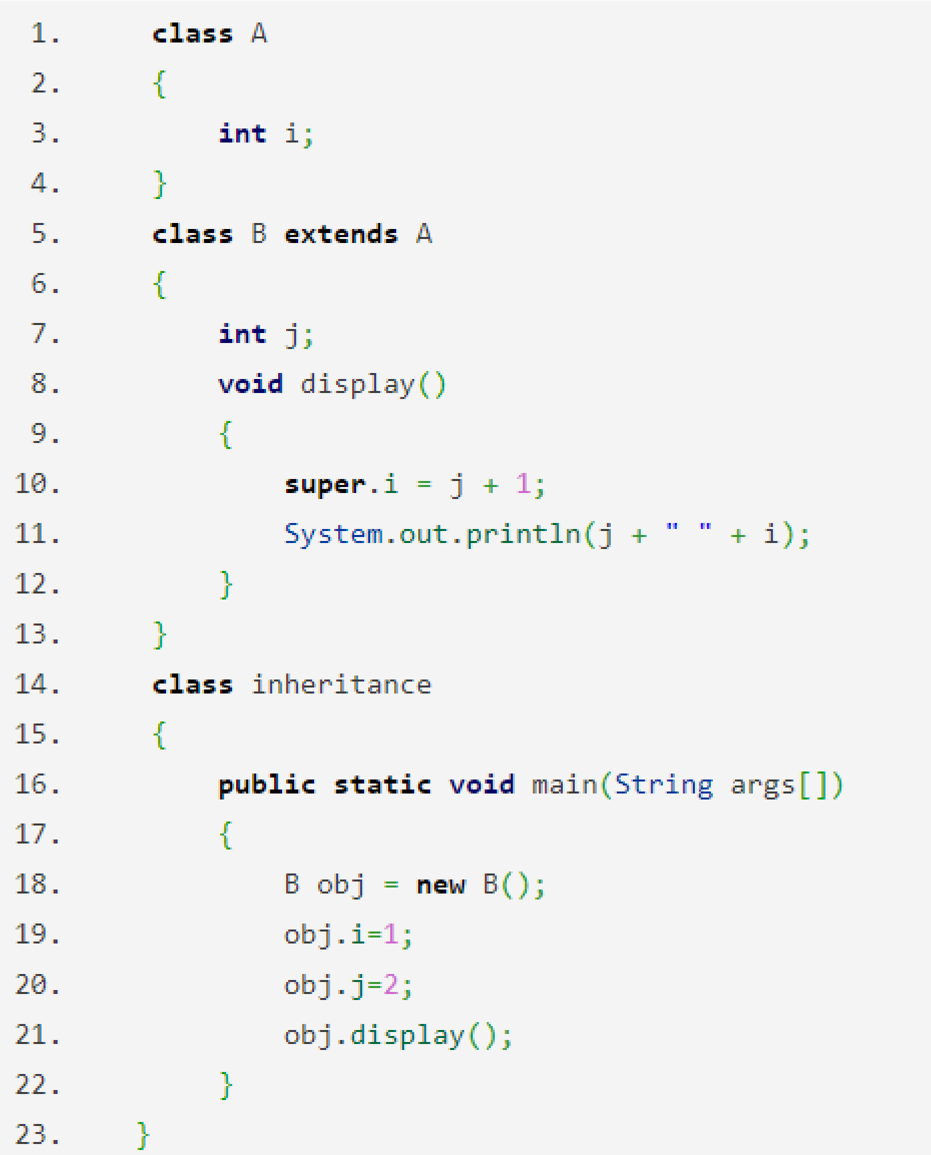
```
class Alpha
{
    static String s = " ";
    protected Alpha()
    {
        s += "alpha ";
    }
}
class SubAlpha extends Alpha
{
    private SubAlpha()
    {
        s += "sub ";
    }
}

public class SubSubAlpha extends Alpha
{
    private SubSubAlpha()
    {
        s += "subsub ";
    }
    public static void main(String[] args)
    {
        new SubSubAlpha();
        System.out.println(s);
    }
}
```

alpha subsub ✗

Correct answer

alpha subsub

✓ 1/1



```
class A
{
    int i;
}
class B extends A
{
    int j;
    void display()
    {
        super.i = j + 1;
        System.out.println(j + " " + i);
    }
}
class inheritance
{
    public static void main(String args[])
    {
        B obj = new B();
        obj.i=1;
        obj.j=2;
        obj.display();
    }
}
```

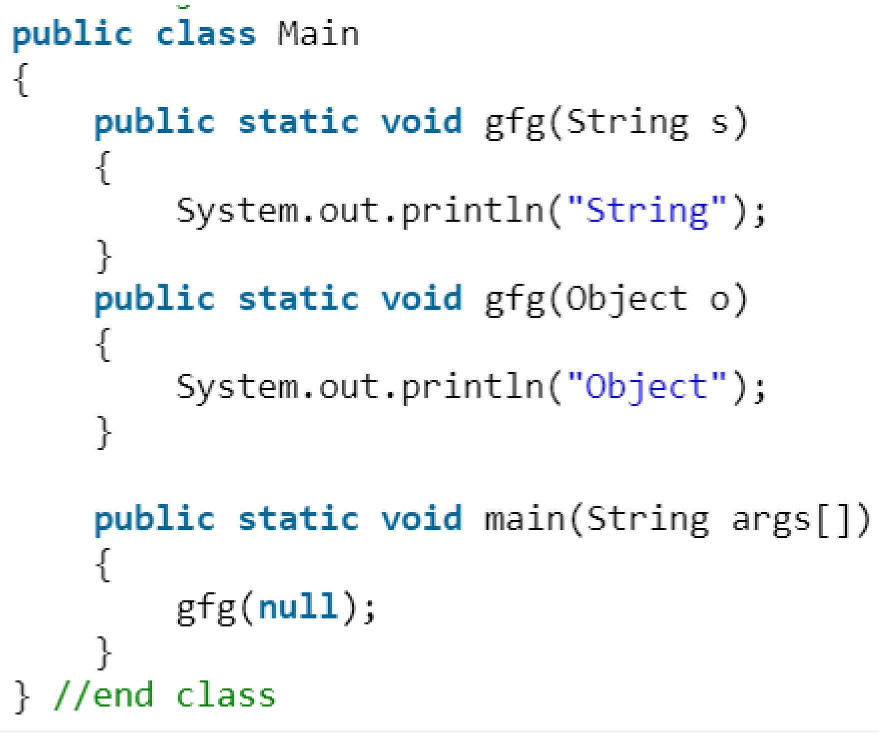
○ a) 2 2

○ b) 3 3

● c) 2 3 ✓

○ d) 3 2

✗ ···/1



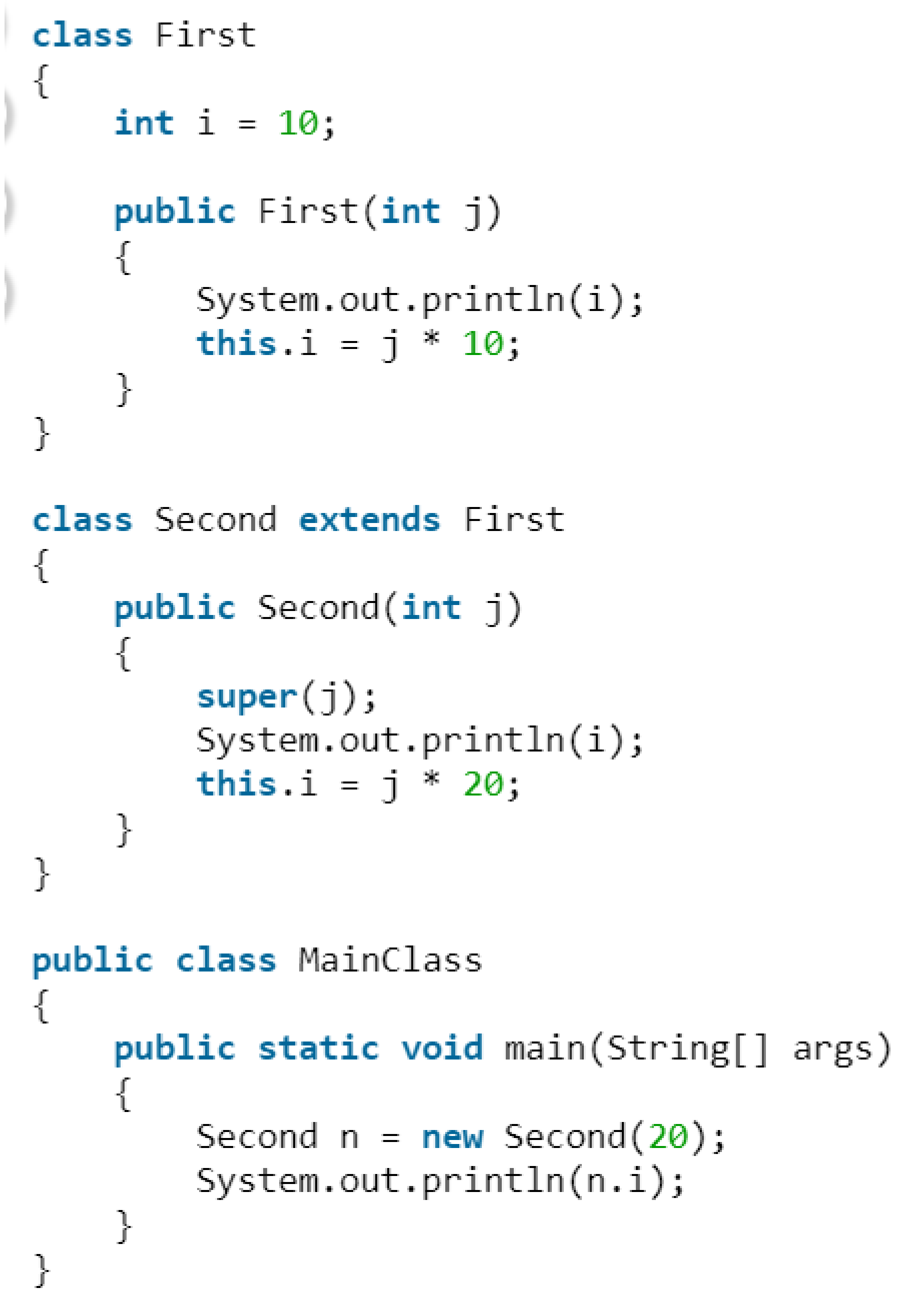
```
public class Main
{
    public static void gfg(String s)
    {
        System.out.println("String");
    }
    public static void gfg(Object o)
    {
        System.out.println("Object");
    }

    public static void main(String args[])
    {
        gfg(null);
    }
} //end class
```

runtime error ✗

Correct answer

String

…/1



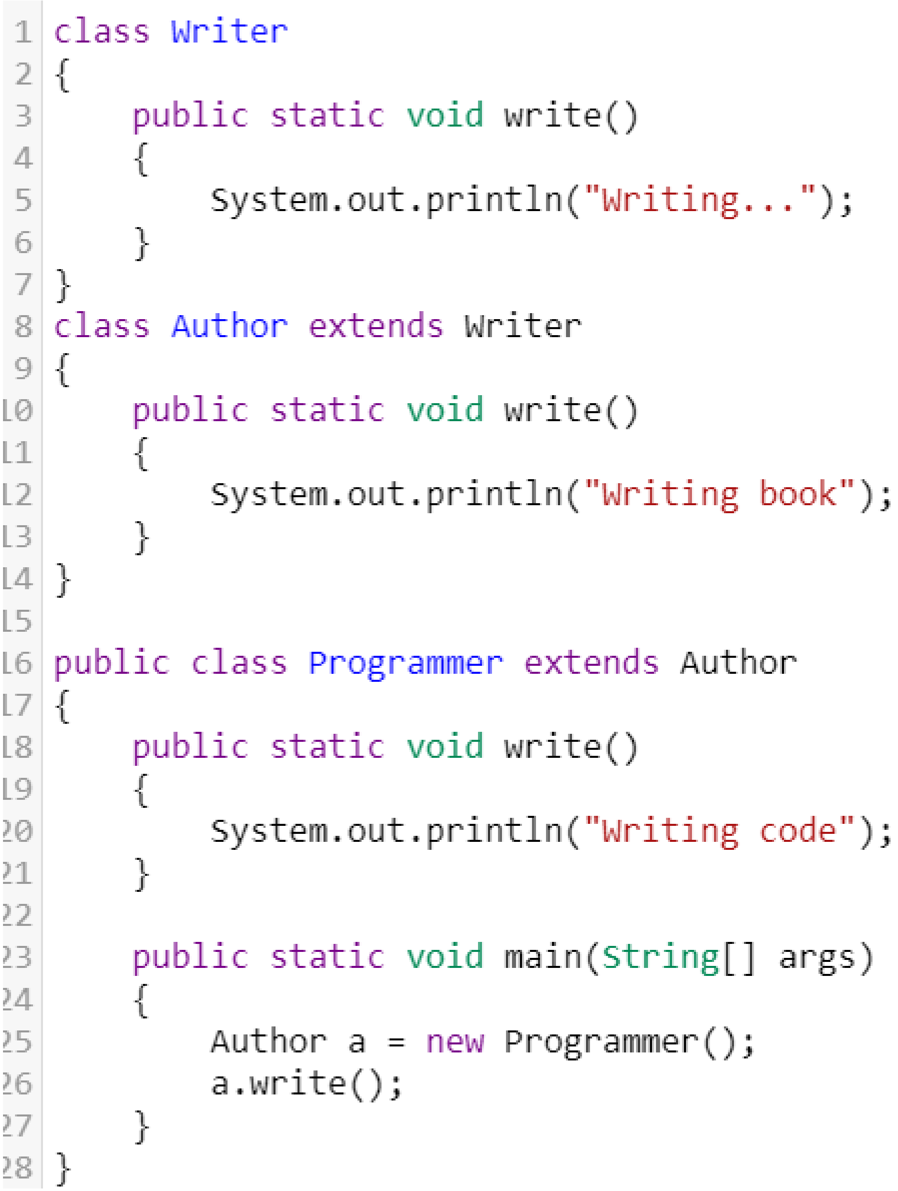
```
class First
{
    int i = 10;

    public First(int j)
    {
        System.out.println(i);
        this.i = j * 10;
    }
}

class Second extends First
{
    public Second(int j)
    {
        super(j);
        System.out.println(i);
        this.i = j * 20;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Second n = new Second(20);
        System.out.println(n.i);
    }
}
```

10 200 400

Correct answer

10 200 400

✗ 0/1



```java
class Writer
{
    public static void write()
    {
        System.out.println("Writing...");
    }
}
class Author extends Writer
{
    public static void write()
    {
        System.out.println("Writing book");
    }
}

public class Programmer extends Author
{
    public static void write()
    {
        System.out.println("Writing code");
    }

    public static void main(String[] args)
    {
        Author a = new Programmer();
        a.write();
    }
}
```

○ A) Writing...

○ B) Writing book

● C) Writing code ✗

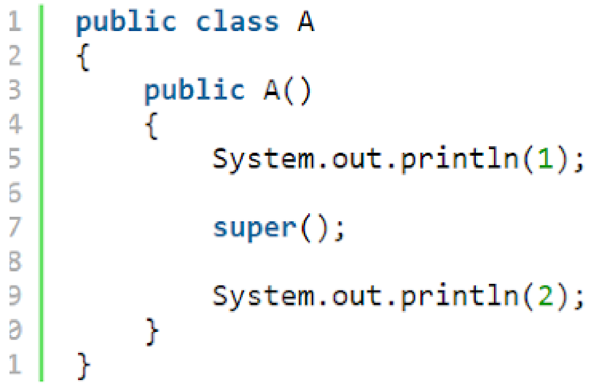
○ D) Compilation fails

Correct answer

● B) Writing book

✗ ···/1

What is wrong with the below code? Why it is showing compile time error?

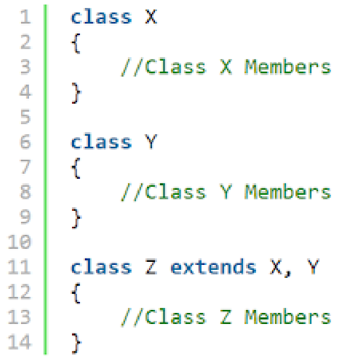


```
public class A
{
    public A()
    {
        System.out.println(1);

        super();

        System.out.println(2);
    }
}
```

no main method ✗

✗ ···/1

Tinku has written the code like below. But, it is showing compile time error. Can you identify what mistake he has done?



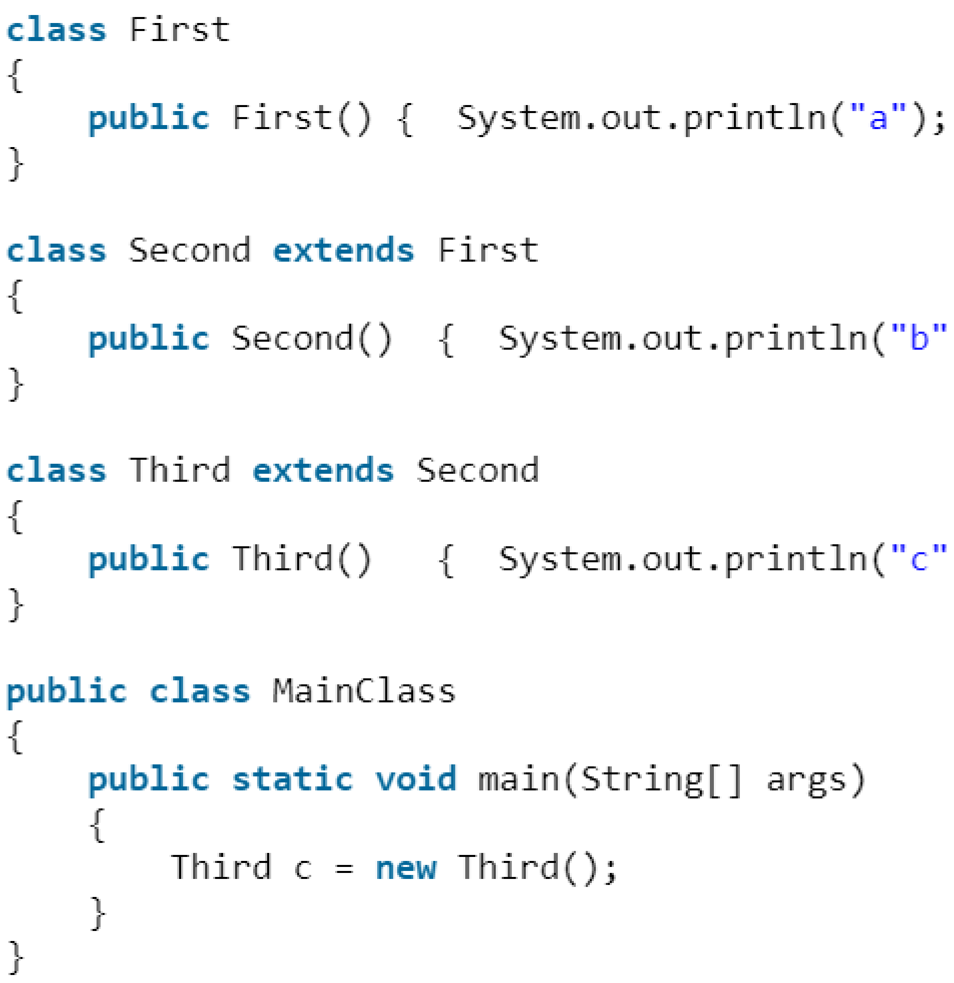
```
class X
{
    //Class X Members
}

class Y
{
    //Class Y Members
}

class Z extends X, Y
{
    //Class Z Members
}
```

multiple inheritance not allowed

Feedback

In Java, a class can not extend more than one class. Class Z is extending two classes – Class X and Class Y. It is a compile time error in java.

✗ …/1



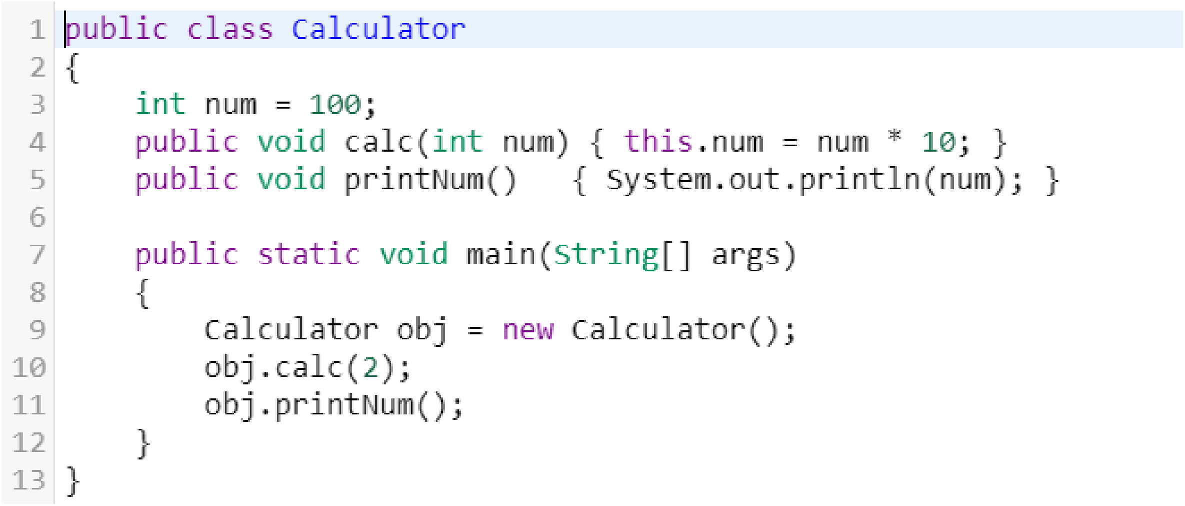
```
class First
{
    public First() {  System.out.println("a");
}

class Second extends First
{
    public Second()  {  System.out.println("b"
}

class Third extends Second
{
    public Third()   {  System.out.println("c"
}

public class MainClass
{
    public static void main(String[] args)
    {
        Third c = new Third();
    }
}
```

abc ✗

Correct answer

a b c

✓ 1/1



```
public class Calculator
{
    int num = 100;
    public void calc(int num) { this.num = num * 10; }
    public void printNum()   { System.out.println(num); }

    public static void main(String[] args)
    {
        Calculator obj = new Calculator();
        obj.calc(2);
        obj.printNum();
    }
}
```

● A) 20 ✓

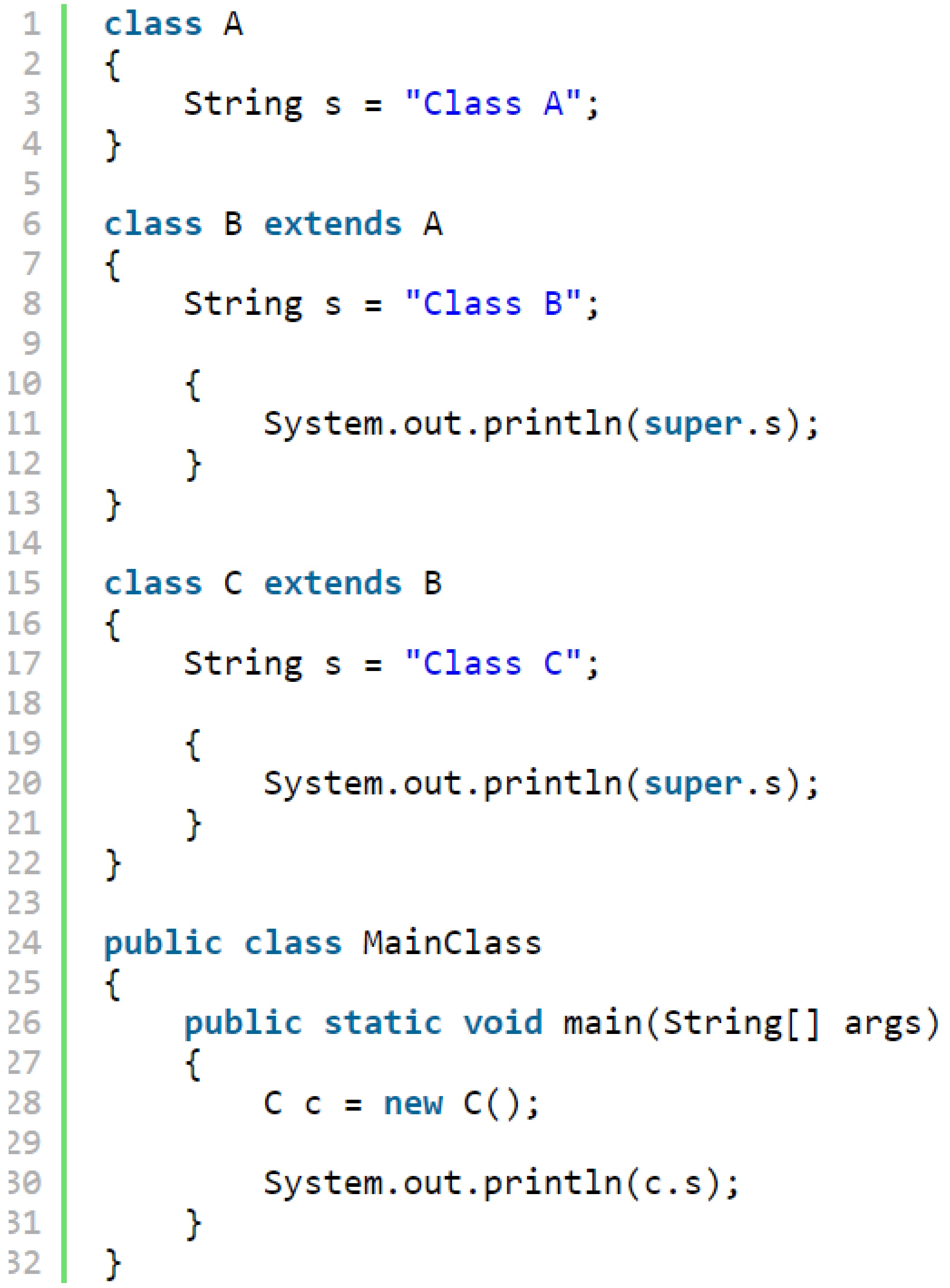
○ B) 100

○ C) 1000

○ D) 2

✗ …/1

What will be the output of the following program?



```
class A
{
    String s = "Class A";
}

class B extends A
{
    String s = "Class B";

    {
        System.out.println(super.s);
    }
}

class C extends B
{
    String s = "Class C";

    {
        System.out.println(super.s);
    }
}

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

        System.out.println(c.s);
    }
}
```

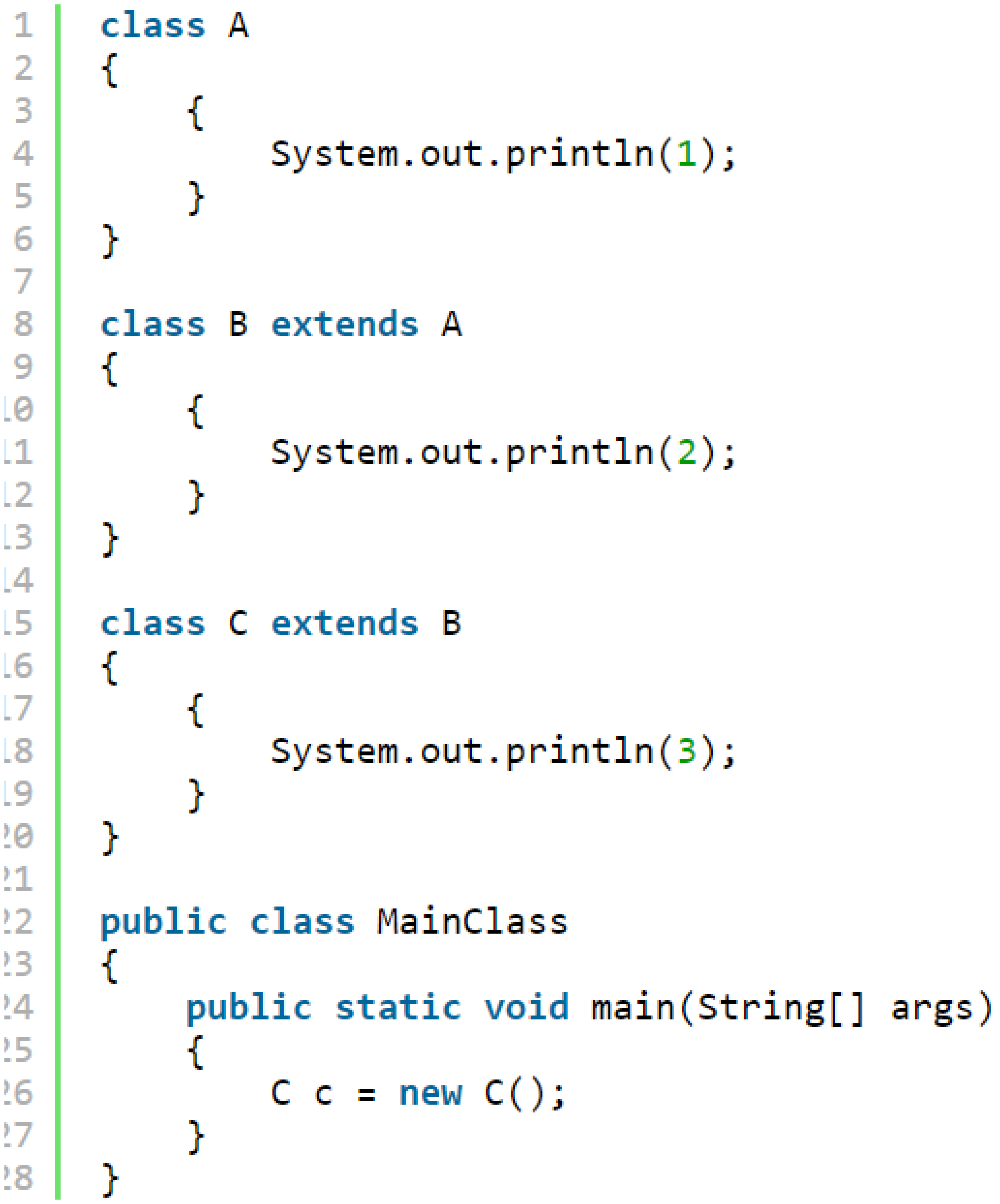
Class A Class b Class C ✗

Correct answer

Class A Class B Class c

✗ …/1

What will be the output of the below program?



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

class B extends A
{
    {
        System.out.println(2);
    }
}

class C extends B
{
    {
        System.out.println(3);
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        C c = new C();
    }
}
```

123 ✗

Correct answer

1 2 3

Student_id *

200240320138