INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

Department of Computer Science and Engineering

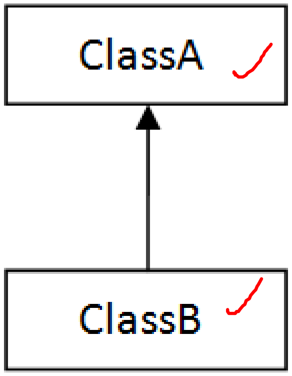
Indian Institute of Technology Roorkee

Roorkee 247 667

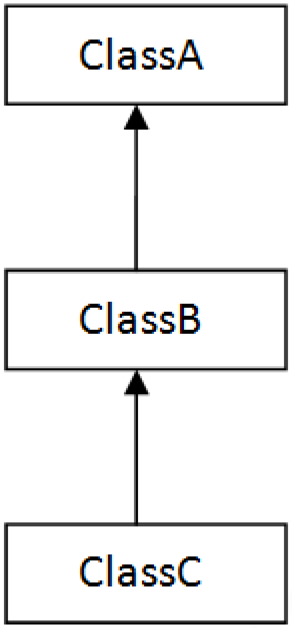
balarfcs@iitr.ac.in

https://sites.google.com/site/balaiitr/

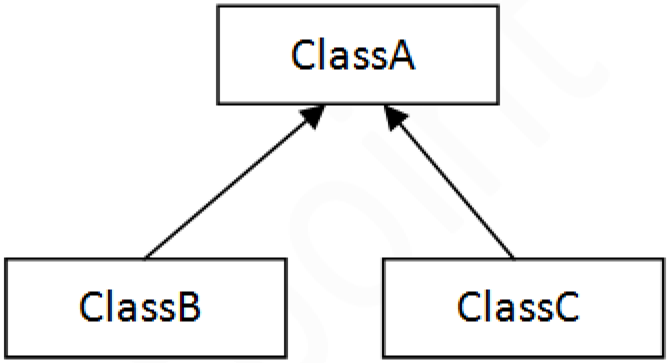
Types of Inheritance



1) Single



2) Multilevel

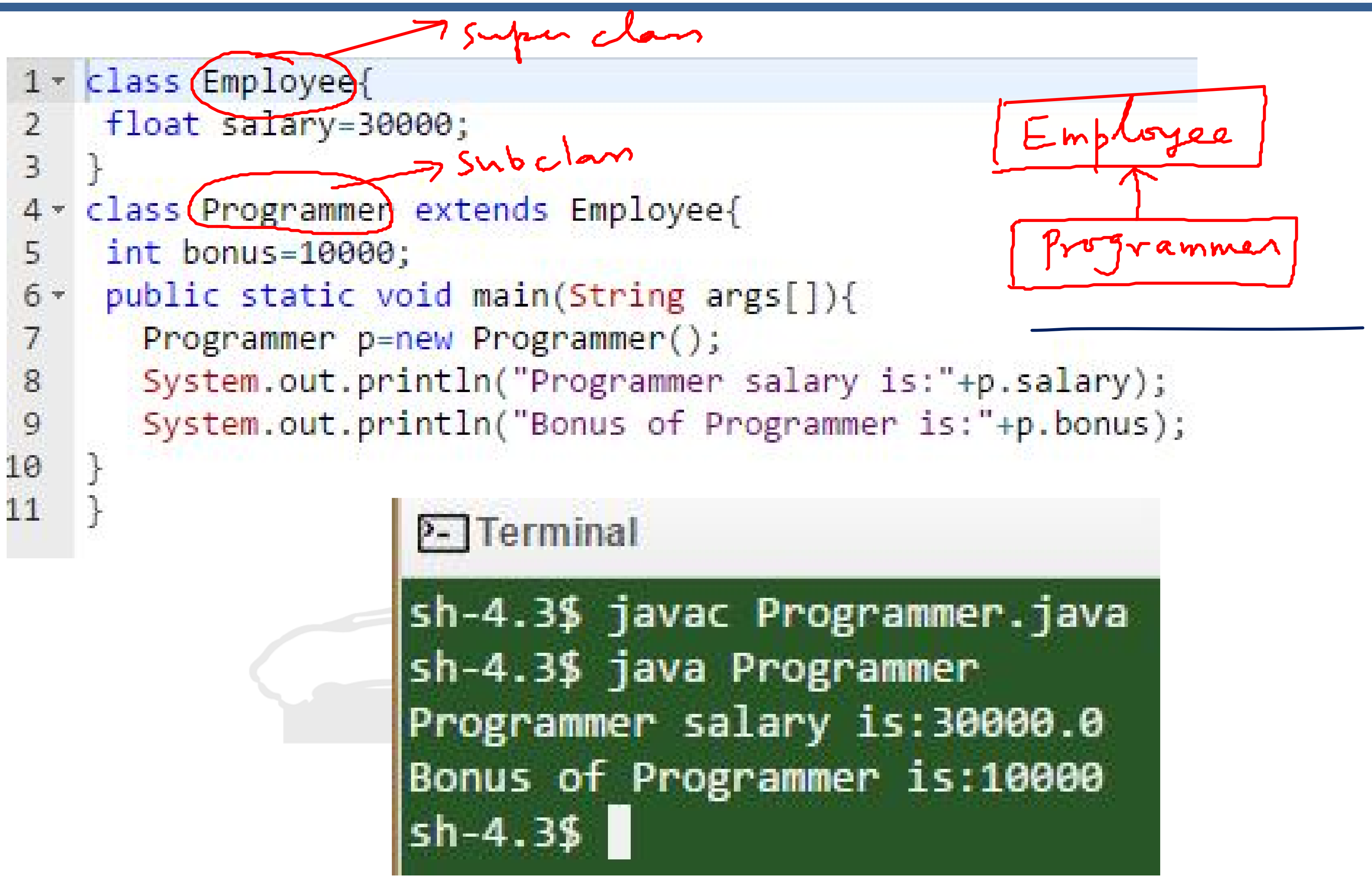


3) Hierarchical

Syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{
   //methods and fields
}
```

Simple or Single Inheritance

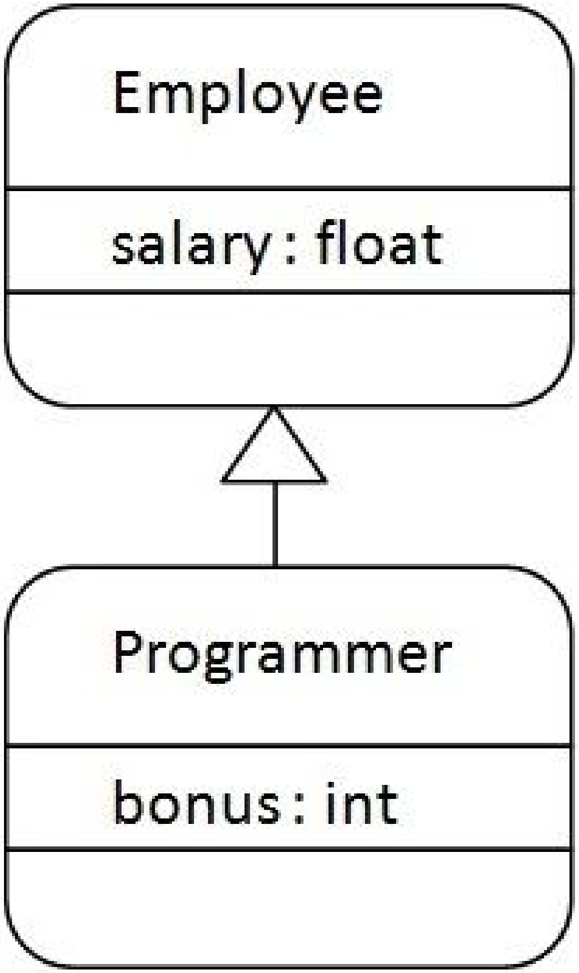


```
class Employee{
 float salary=30000;
 }
class Programmer extends Employee{
 int bonus=10000;
 public static void main(String args[]){
   Programmer p=new Programmer();
   System.out.println("Programmer salary is:"+p.salary);
   System.out.println("Bonus of Programmer is:"+p.bonus);
 }
 }
```

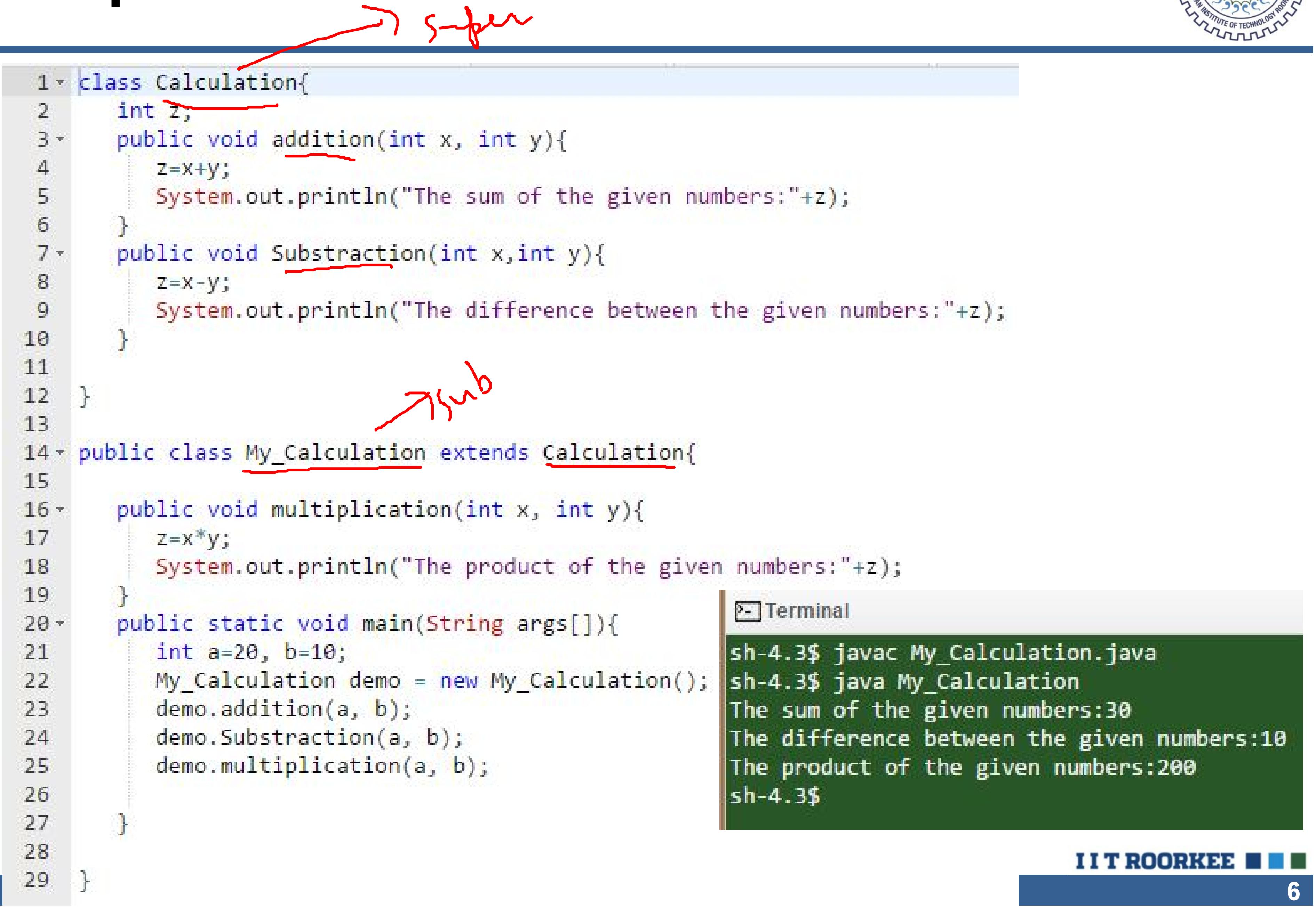
Terminal

```
sh-4.3$ javac Programmer.java
sh-4.3$ java Programmer
Programmer salary is:30000.0
Bonus of Programmer is:10000
sh-4.3$
```

Understanding Simple Inheritance



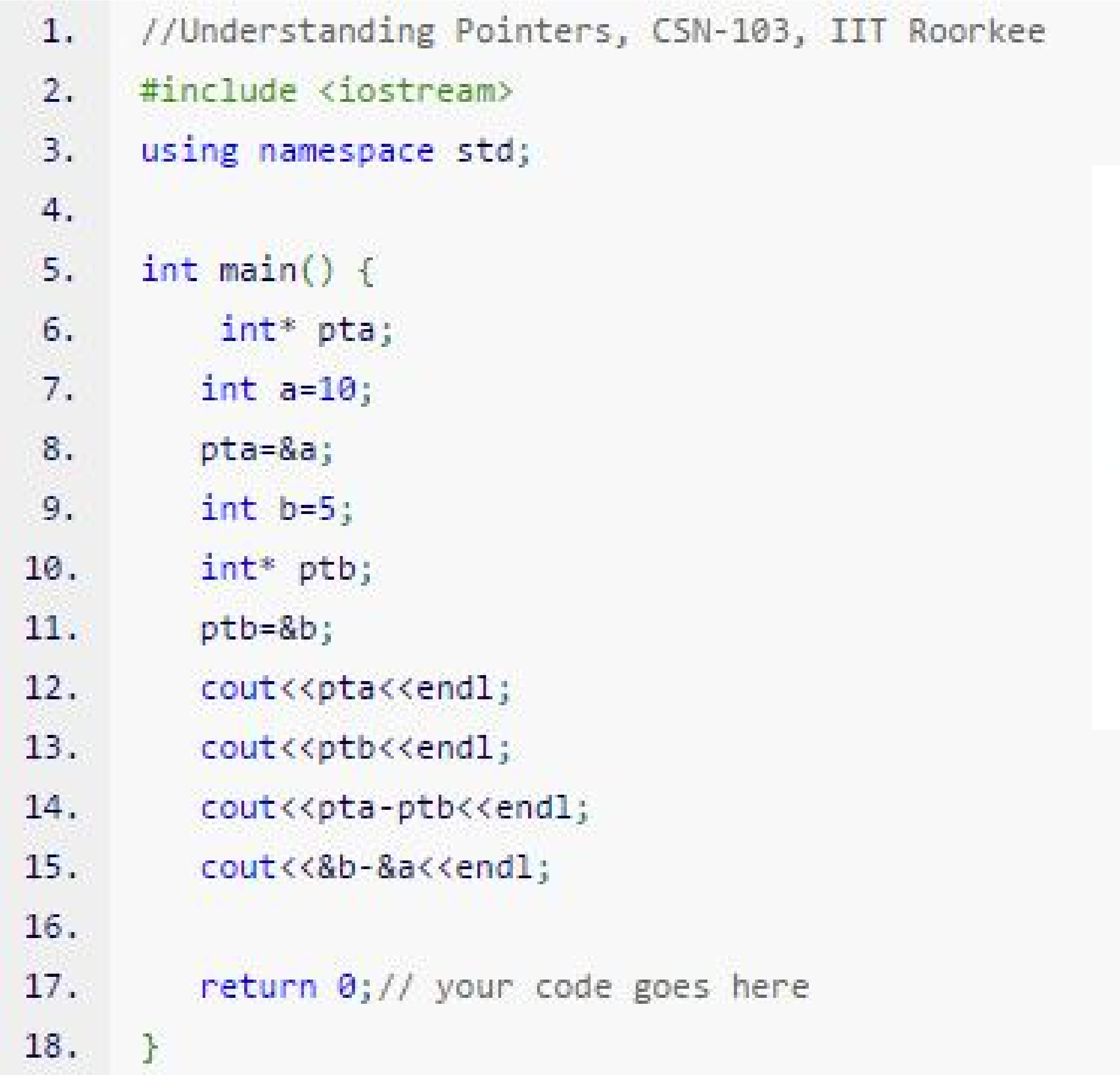
Simple Inheritance



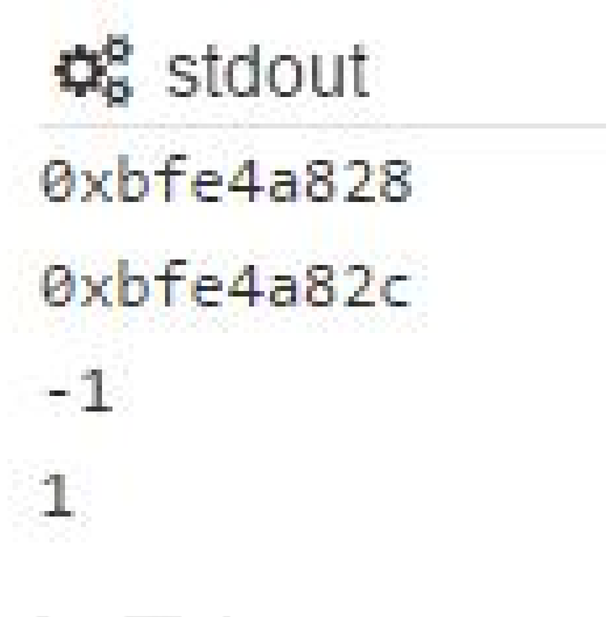
```
class Calculation{
    int z;
    public void addition(int x, int y){
        z=x+y;
        System.out.println("The sum of the given numbers:"+z);
    }
    public void Substraction(int x,int y){
        z=x-y;
        System.out.println("The difference between the given numbers:"+z);
    }

}

public class My_Calculation extends Calculation{

    public void multiplication(int x, int y){
        z=x*y;
        System.out.println("The product of the given numbers:"+z);
    }
    public static void main(String args[]){
        int a=20, b=10;
        My_Calculation demo = new My_Calculation();
        demo.addition(a, b);
        demo.Substraction(a, b);
        demo.multiplication(a, b);

    }

}
```

```
sh-4.3$ javac My_Calculation.java
sh-4.3$ java My_Calculation
The sum of the given numbers:30
The difference between the given numbers:10
The product of the given numbers:200
sh-4.3$
```

Understanding Pointers

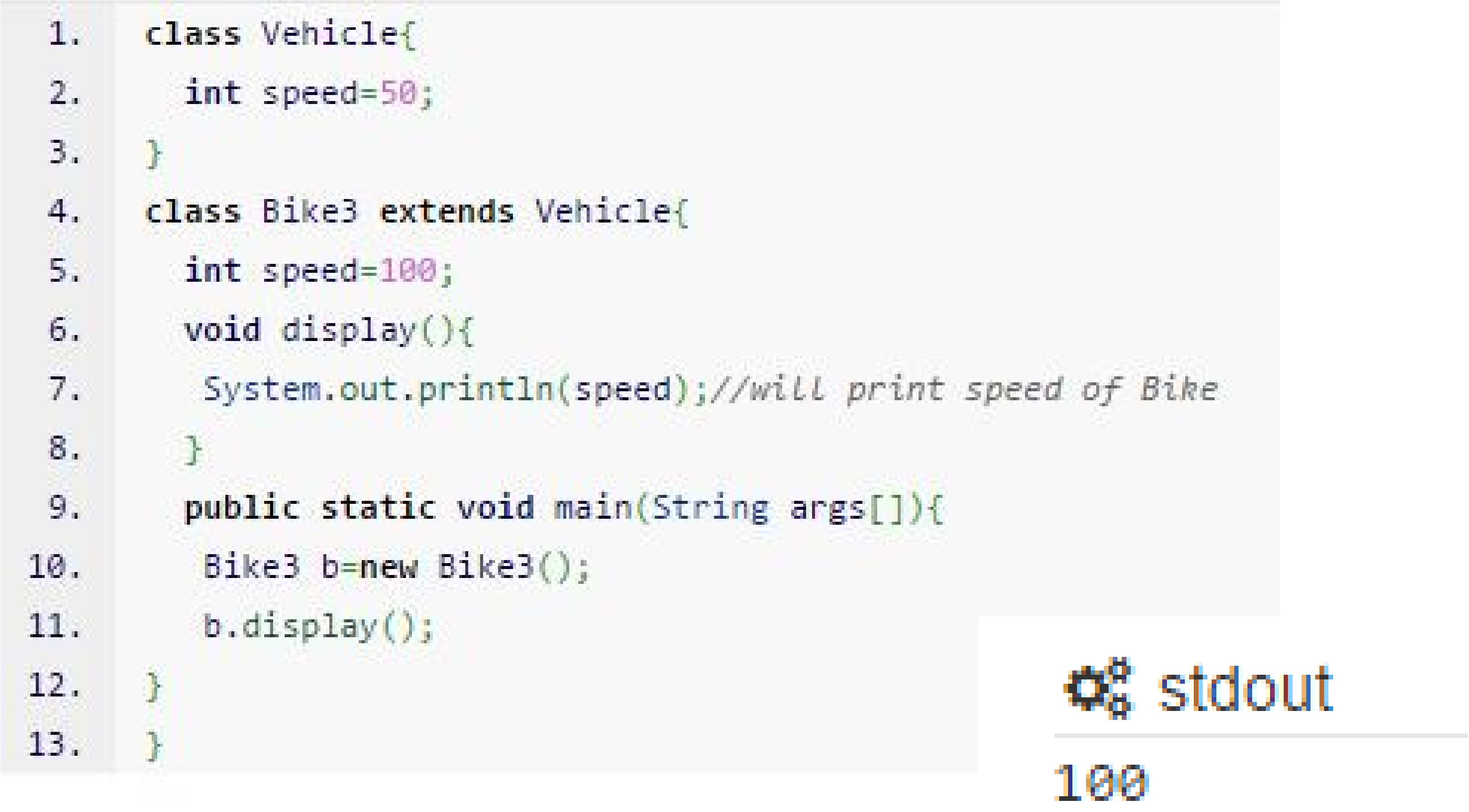


```
//Understanding Pointers, CSN-103, IIT Roorkee
#include <iostream>
using namespace std;

int main() {
     int* pta;
    int a=10;
    pta=&a;
    int b=5;
    int* ptb;
    ptb=&b;
    cout<<pta<<endl;
    cout<<ptb<<endl;
    cout<<pta-ptb<<endl;
    cout<<&b-&a<<endl;

    return 0;// your code goes here
}
```



```
0xbfe4a828
0xbfe4a82c
-1
1
```

super keyword in JAVA

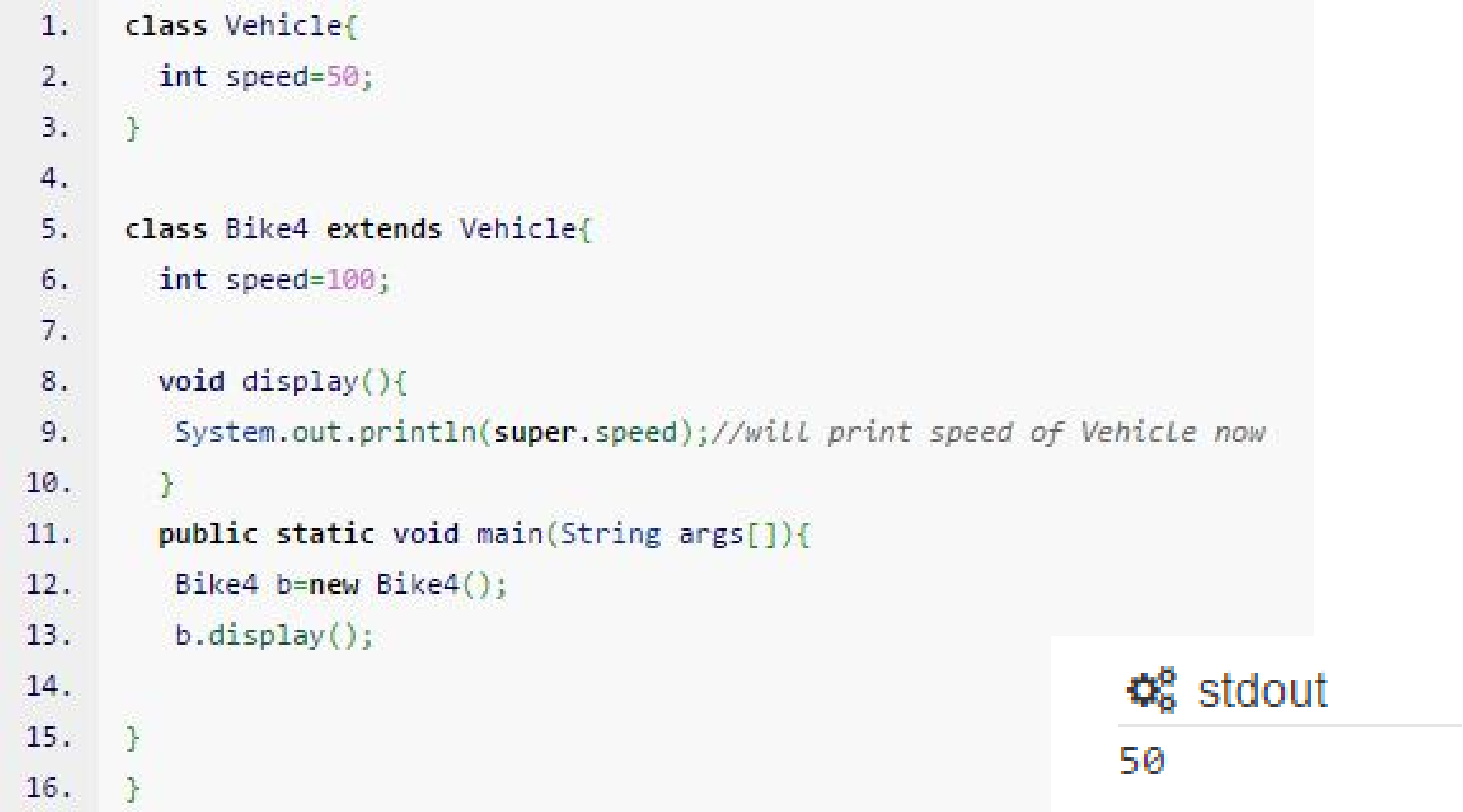


```
class Vehicle{
  int speed=50;
}
class Bike3 extends Vehicle{
  int speed=100;
  void display(){
   System.out.println(speed);//will print speed of Bike
  }
  public static void main(String args[]){
   Bike3 b=new Bike3();
   b.display();
}
}
```

stdout

```
100
```

- https://ideone.com/xi2oPz



```java
class Vehicle{
  int speed=50;
}

class Bike4 extends Vehicle{
  int speed=100;

  void display(){
   System.out.println(super.speed);//will print speed of Vehicle now
  }
  public static void main(String args[]){
   Bike4 b=new Bike4();
   b.display();

}
}
```

```
50
```

- https://ideone.com/yWPiOV

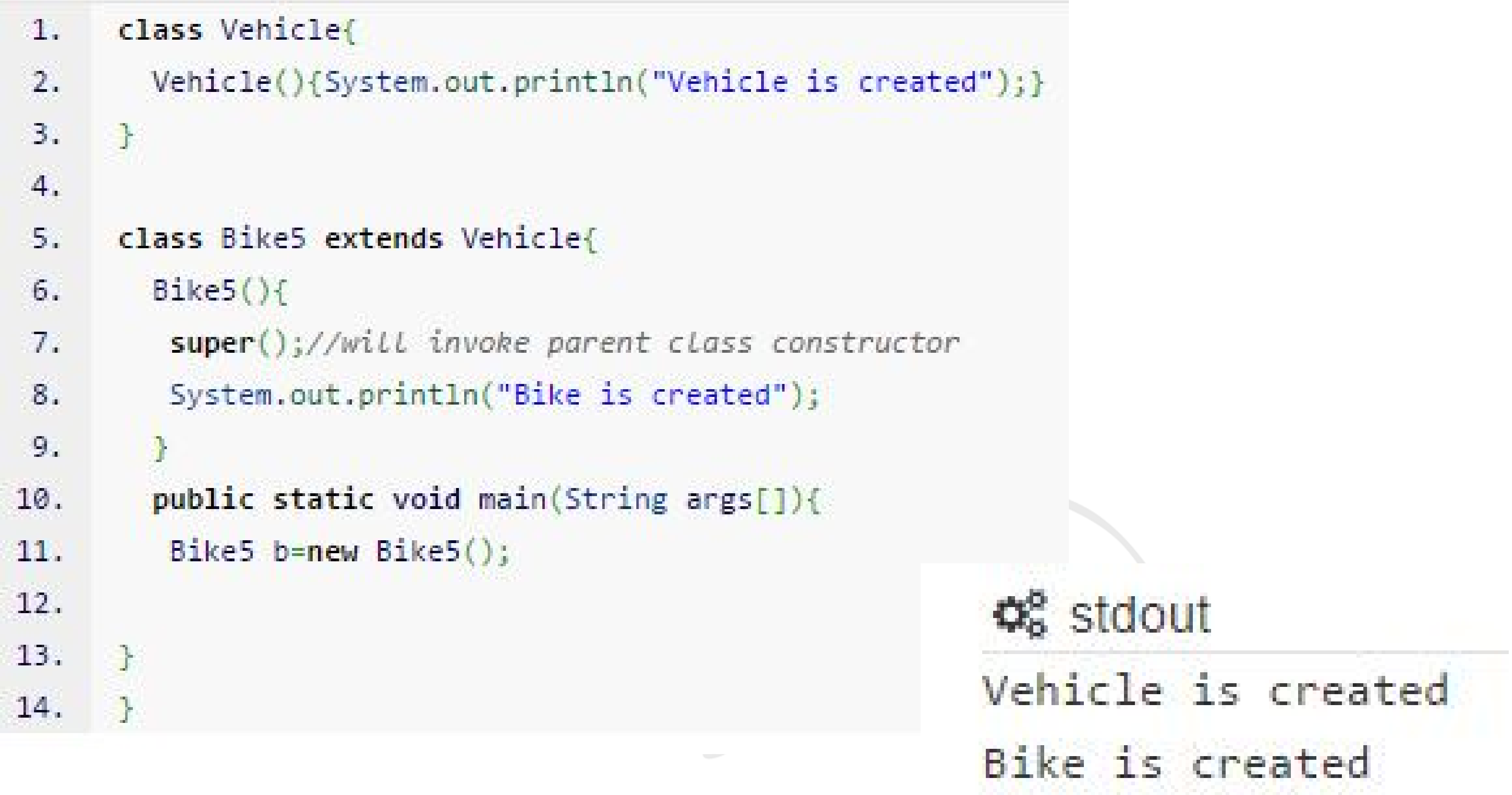
super keyword in java

- The **super** keyword in java is a reference variable that is used to refer immediate parent class object.

- Whenever you create the instance of subclass, an instance of parent class is created implicitly i.e. referred by super reference variable.

Usage of java super Keyword

- super is used to refer immediate parent class instance variable.
- super() is used to invoke immediate parent class constructor.
- super is used to invoke immediate parent class method.

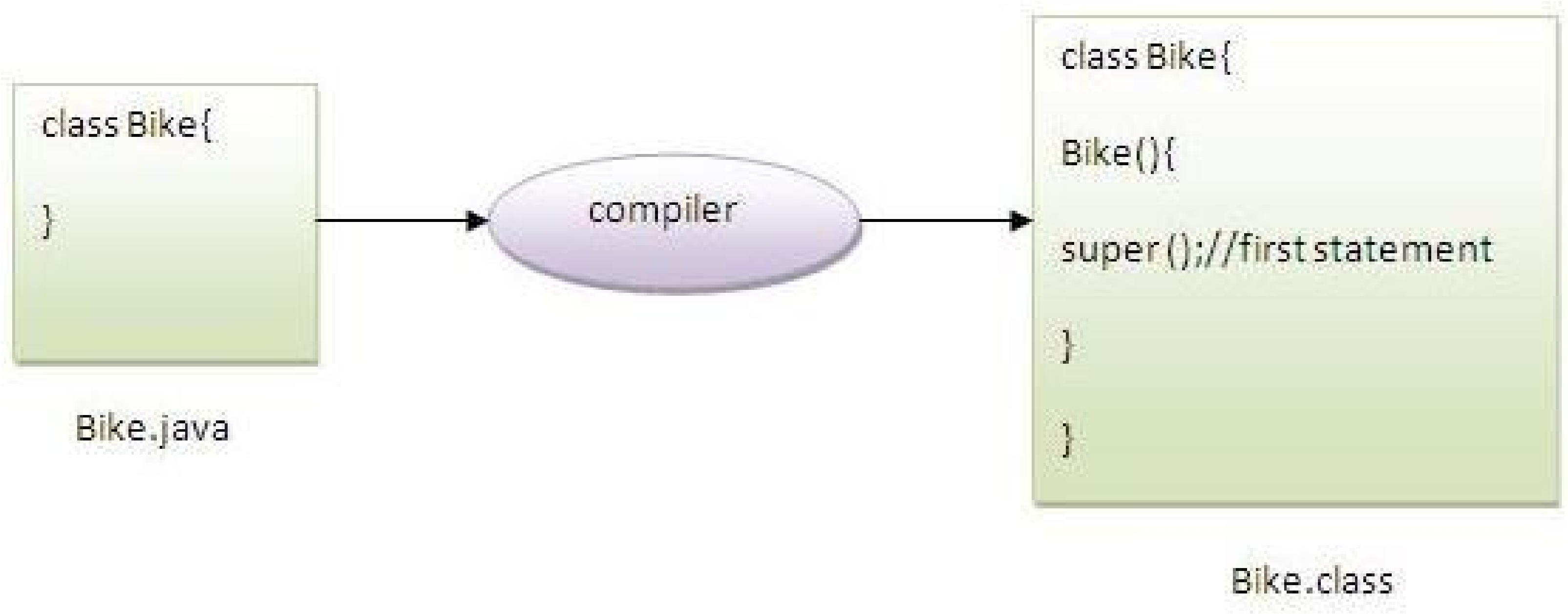


```java
class Vehicle{
  Vehicle(){System.out.println("Vehicle is created");}
}

class Bike5 extends Vehicle{
  Bike5(){
   super();//will invoke parent class constructor
   System.out.println("Bike is created");
  }
  public static void main(String args[]){
   Bike5 b=new Bike5();

}
}
```

```
Vehicle is created
Bike is created
```

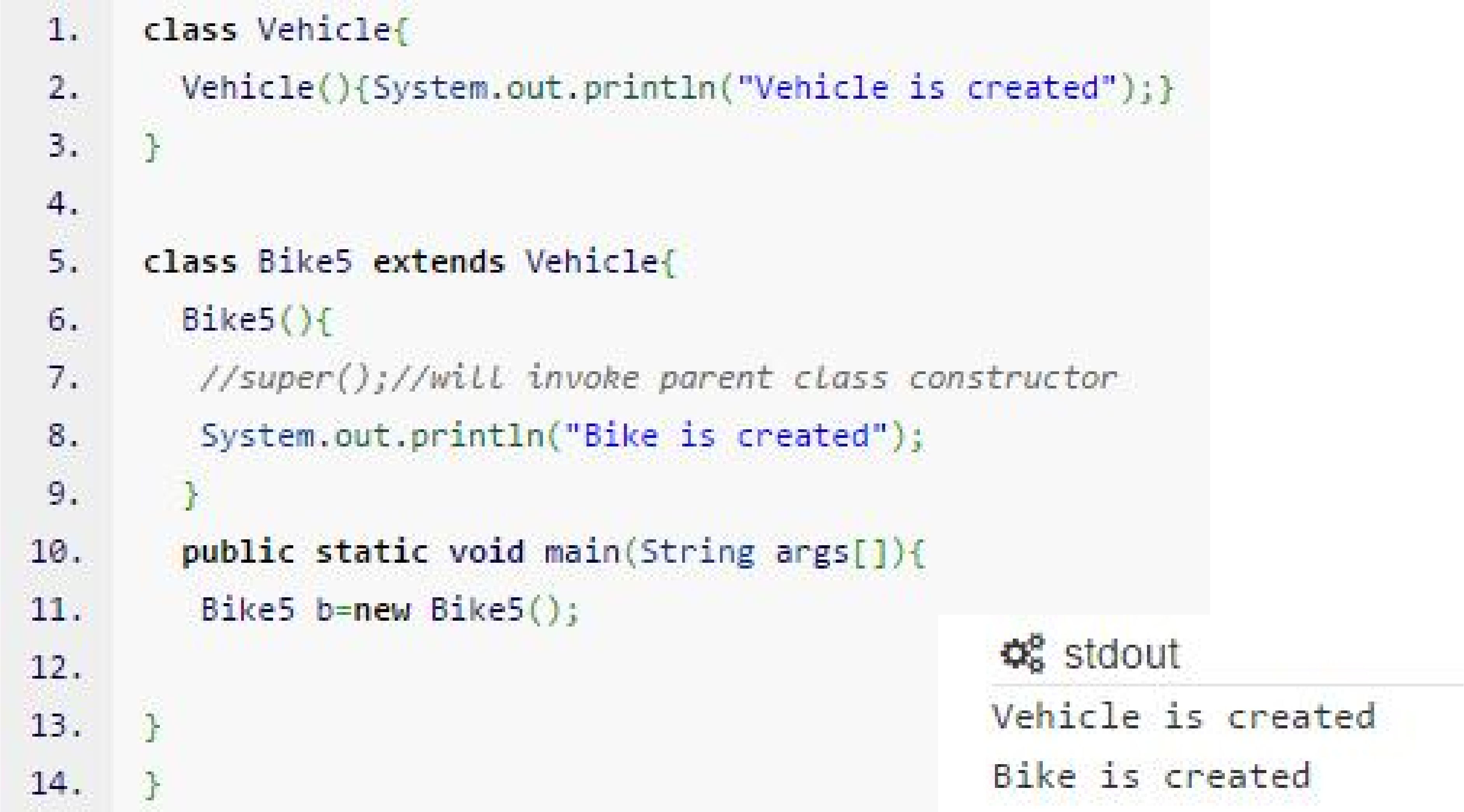
- https://ideone.com/qx3zTd

- super() is added in each class constructor automatically by compiler.



```
class Bike{

}
```

```
class Bike{

Bike(){

super();//first statement

}

}
```

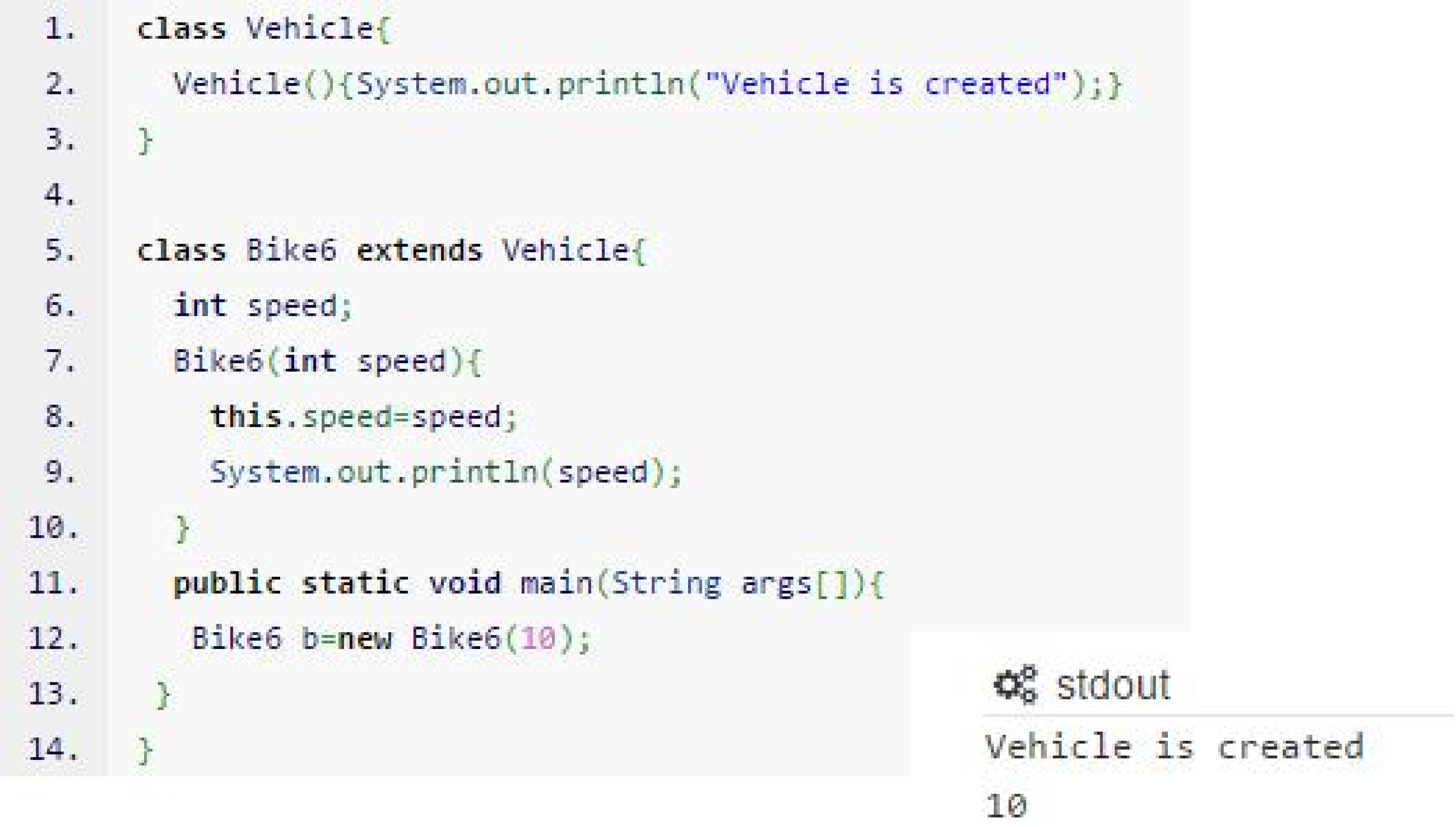


```java
class Vehicle{
  Vehicle(){System.out.println("Vehicle is created");}
}

class Bike5 extends Vehicle{
  Bike5(){
   //super();//will invoke parent class constructor
   System.out.println("Bike is created");
  }
  public static void main(String args[]){
   Bike5 b=new Bike5();

}
}
```

```
Vehicle is created
Bike is created
```

- https://ideone.com/uu87Hk

Another example

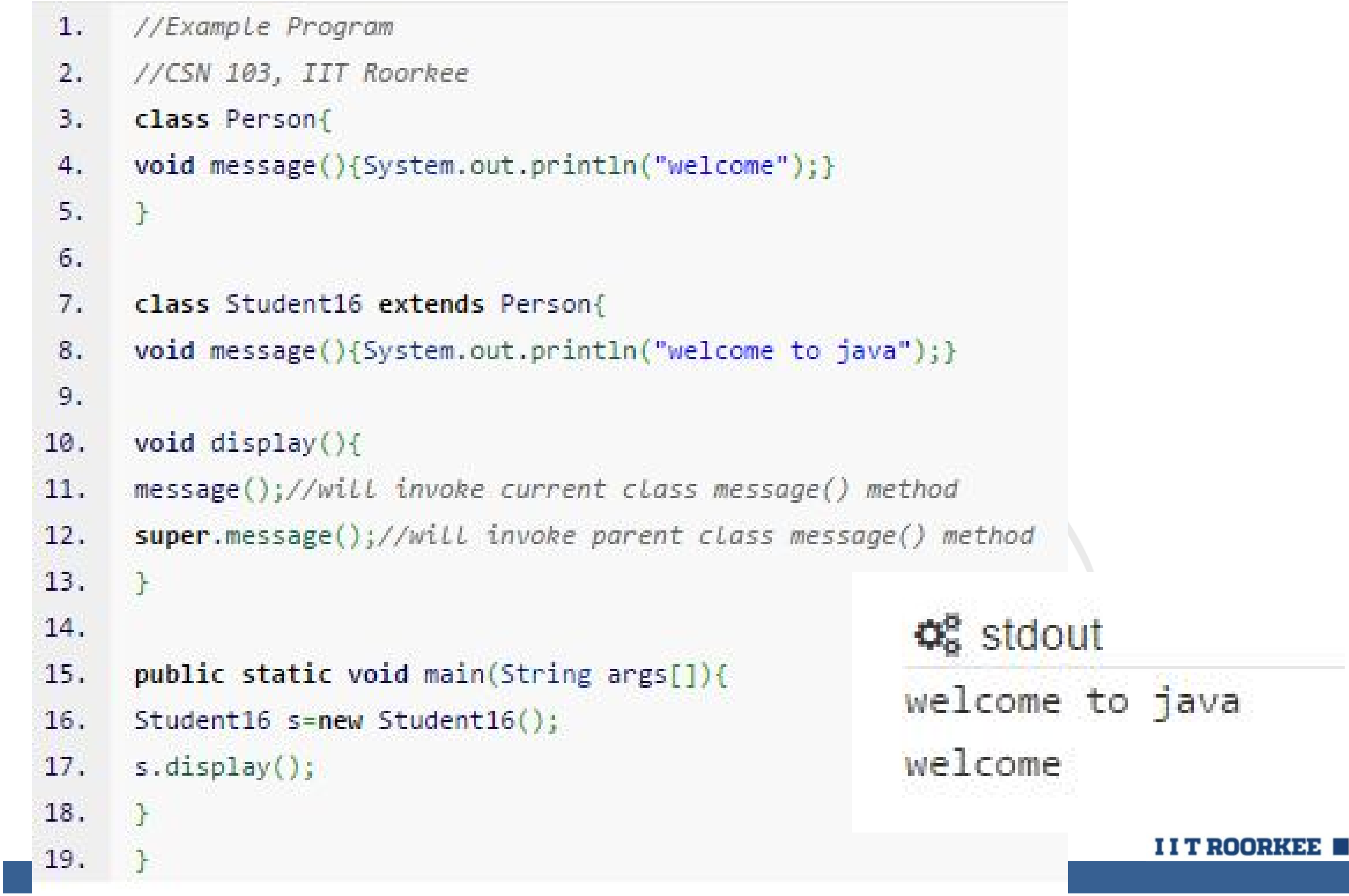


```java
class Vehicle{
  Vehicle(){System.out.println("Vehicle is created");}
}

class Bike6 extends Vehicle{
  int speed;
  Bike6(int speed){
    this.speed=speed;
    System.out.println(speed);
  }
  public static void main(String args[]){
   Bike6 b=new Bike6(10);
 }
}
```

```
Vehicle is created
10
```

- https://ideone.com/EBrcDW

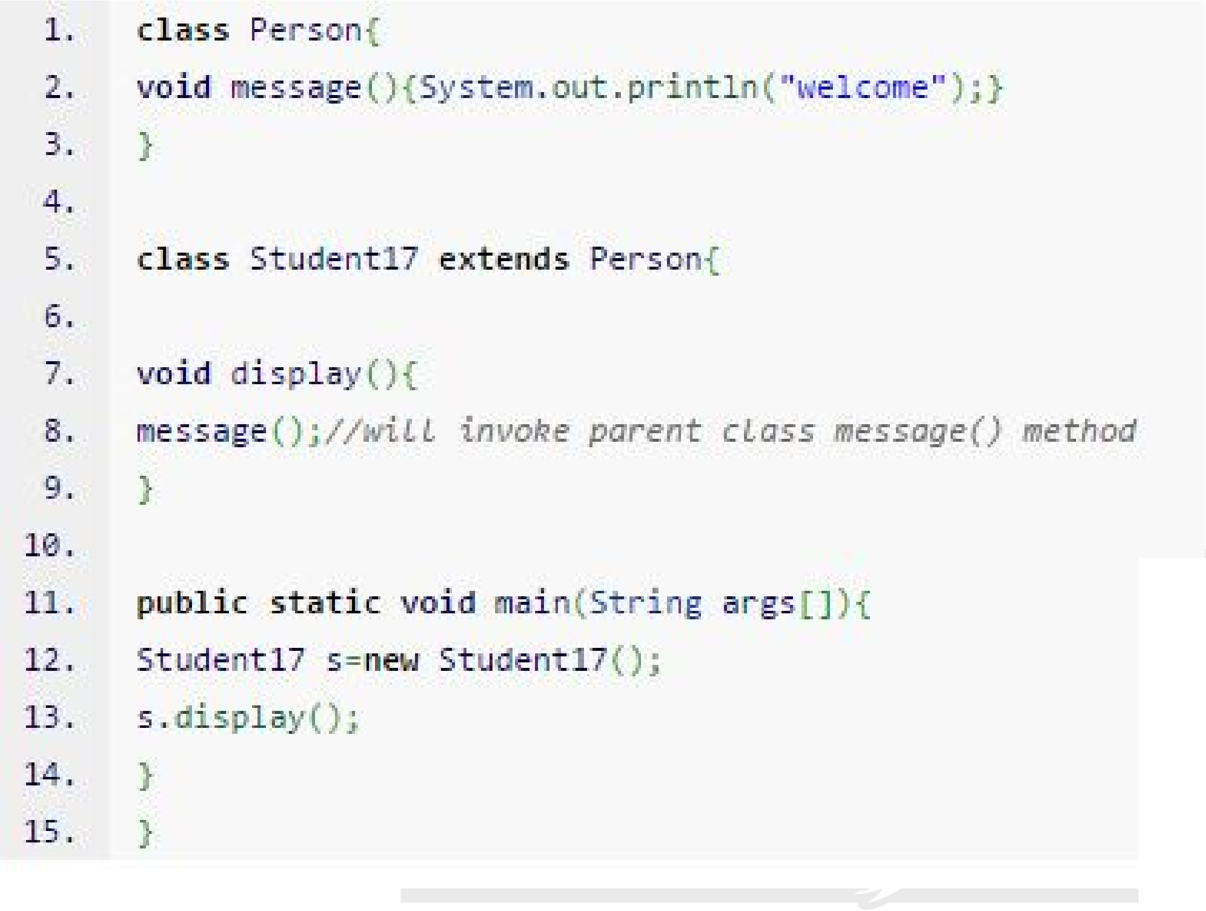
super can be used to invoke parent class method



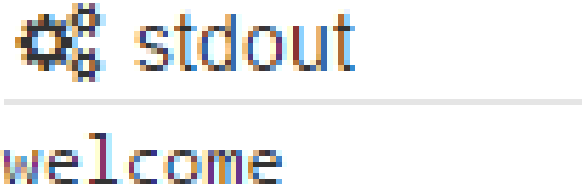
```
//Example Program
//CSN 103, IIT Roorkee
class Person{
void message(){System.out.println("welcome");}
}

class Student16 extends Person{
void message(){System.out.println("welcome to java");}

void display(){
message();//will invoke current class message() method
super.message();//will invoke parent class message() method
}

public static void main(String args[]){
Student16 s=new Student16();
s.display();
}
}
```

```
welcome to java
welcome
```

Program in case super is not required

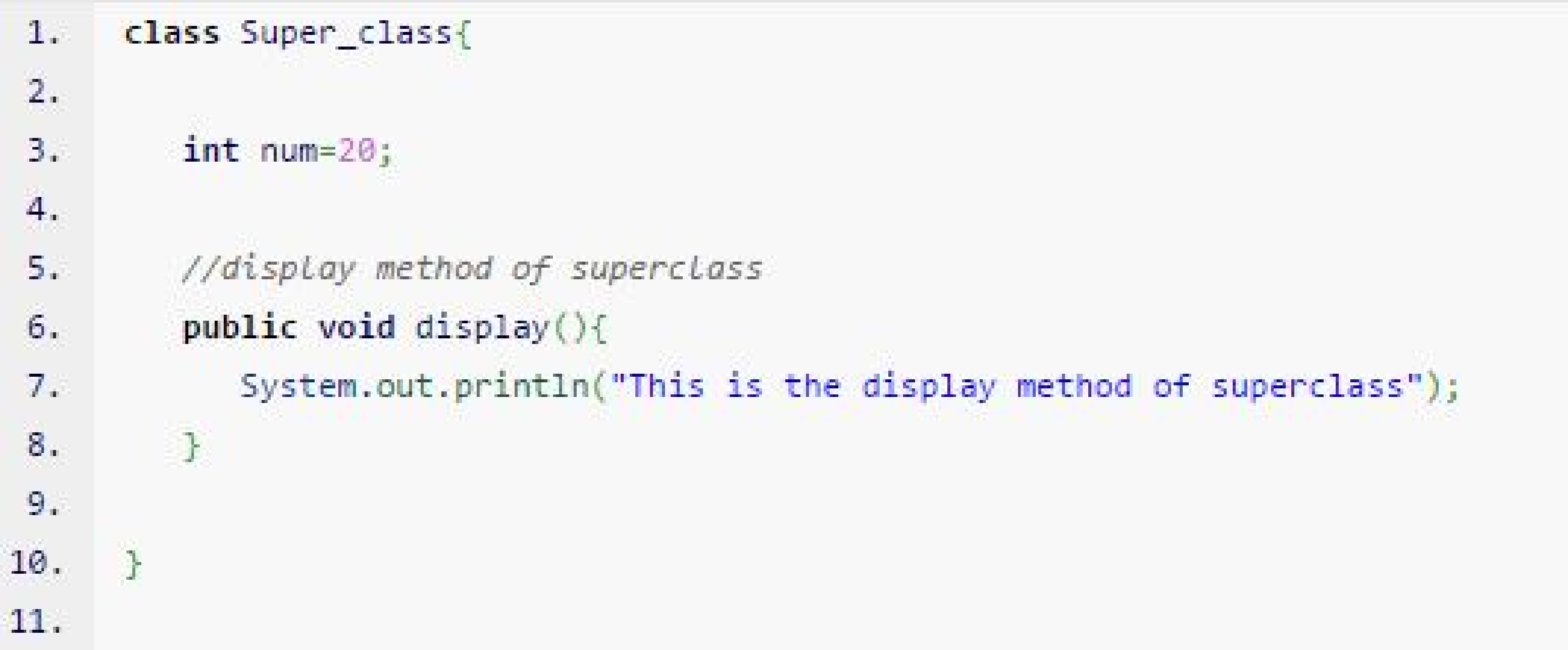


```java
class Person{
void message(){System.out.println("welcome");}
}

class Student17 extends Person{

void display(){
message();//will invoke parent class message() method
}

public static void main(String args[]){
Student17 s=new Student17();
s.display();
}
}
```

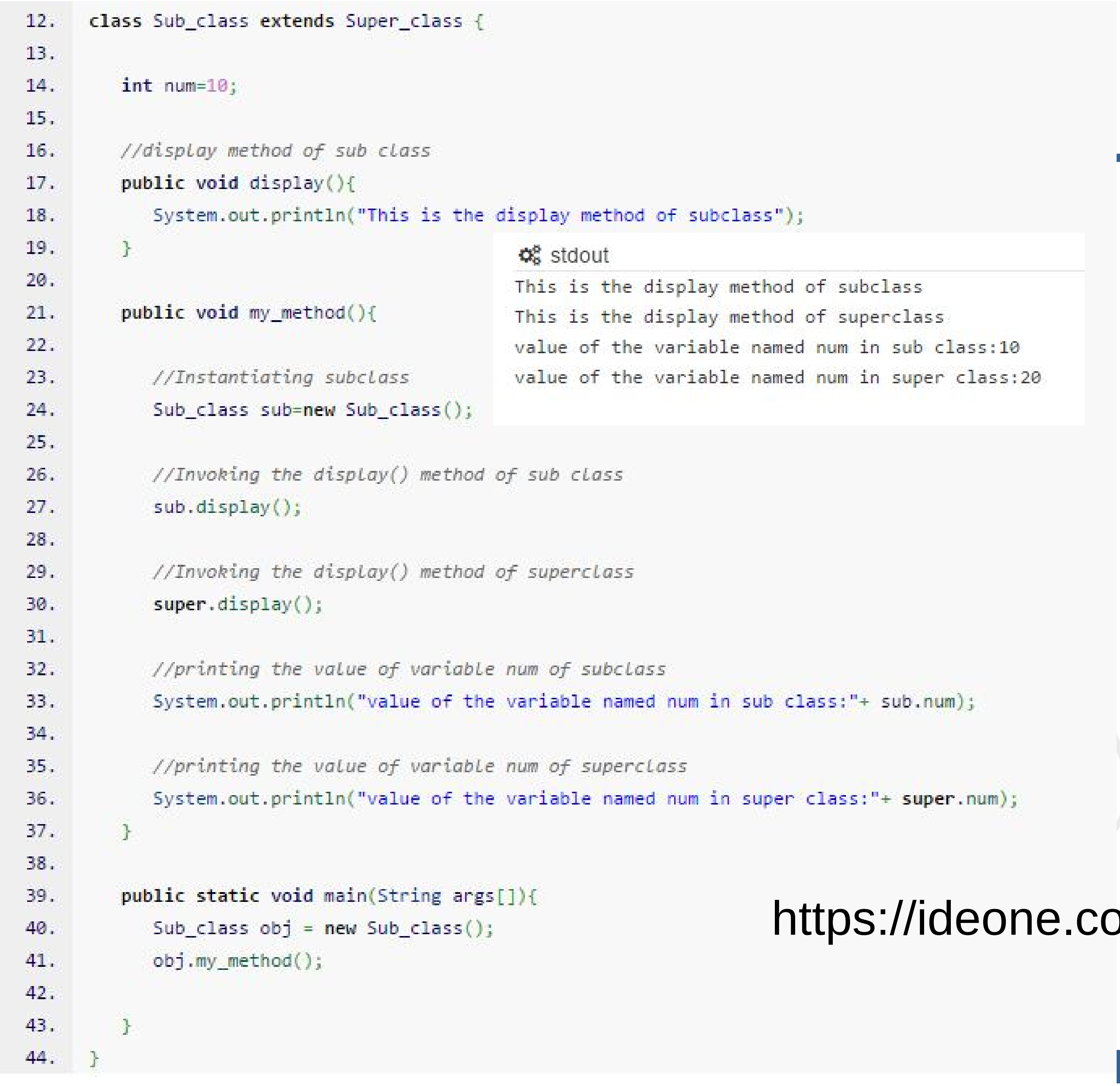


```
welcome
```

- https://ideone.com/idqe3T



```
class Super_class{

  int num=20;

  //display method of superclass
  public void display(){
    System.out.println("This is the display method of superclass");
  }

}

```

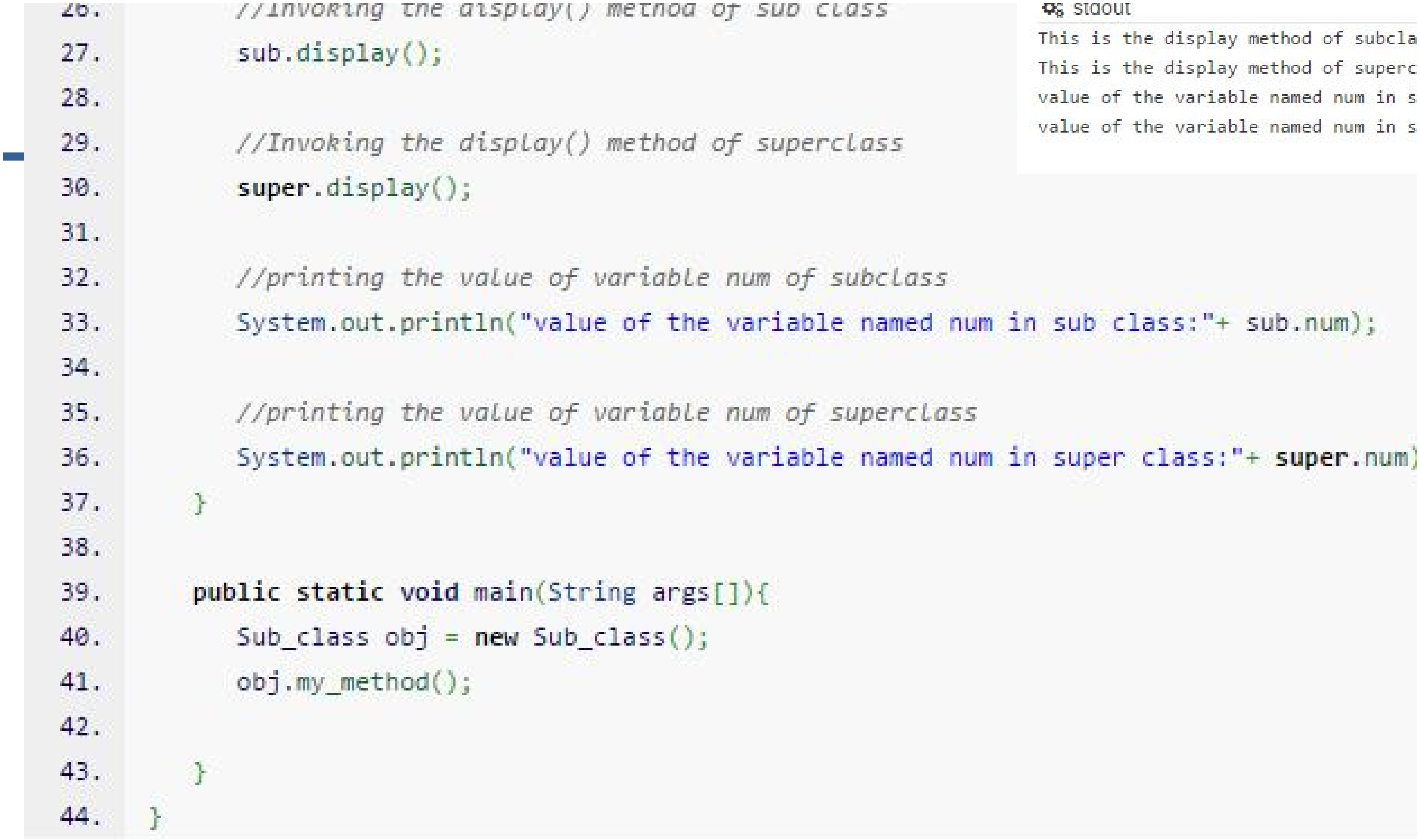


```java
class Sub_class extends Super_class {

   int num=10;

   //display method of sub class
   public void display(){
      System.out.println("This is the display method of subclass");
   }

   public void my_method(){

      //Instantiating subclass
      Sub_class sub=new Sub_class();

      //Invoking the display() method of sub class
      sub.display();

      //Invoking the display() method of superclass
      super.display();

      //printing the value of variable num of subclass
      System.out.println("value of the variable named num in sub class:"+ sub.num);

      //printing the value of variable num of superclass
      System.out.println("value of the variable named num in super class:"+ super.num);
   }

   public static void main(String args[]){
      Sub_class obj = new Sub_class();
      obj.my_method();

   }
}
```

stdout

```
This is the display method of subclass
This is the display method of superclass
value of the variable named num in sub class:10
value of the variable named num in super class:20
```

https://ideone.com/6cVLNg

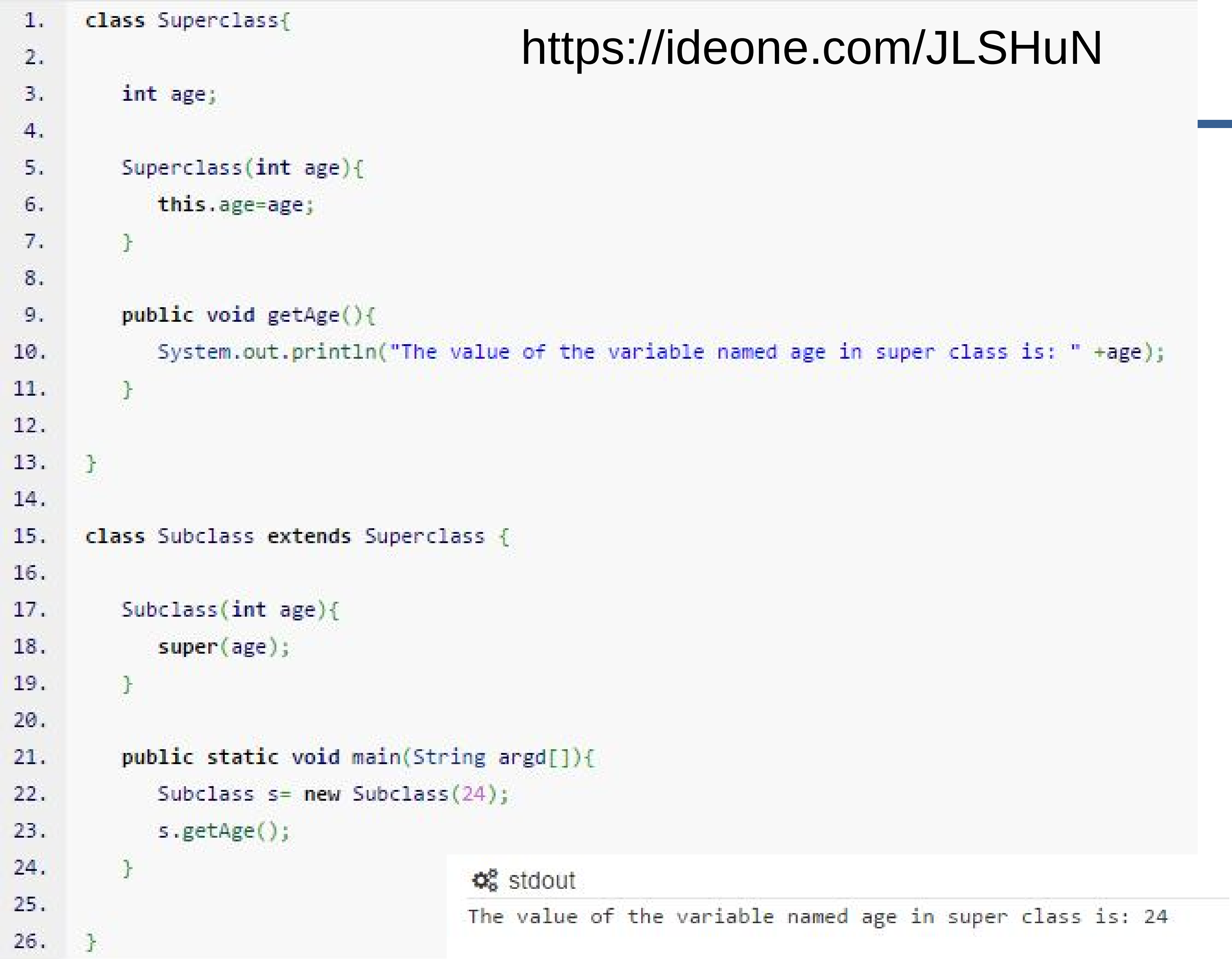


```
      //Invoking the display() method of sub class
      sub.display();

      //Invoking the display() method of superclass
      super.display();

      //printing the value of variable num of subclass
      System.out.println("value of the variable named num in sub class:"+ sub.num);

      //printing the value of variable num of superclass
      System.out.println("value of the variable named num in super class:"+ super.num)
   }

   public static void main(String args[]){
      Sub_class obj = new Sub_class();
      obj.my_method();

   }
}
```

```
This is the display method of subcla
This is the display method of superc
value of the variable named num in s
value of the variable named num in s
```

https://ideone.com/6cVLNg

https://ideone.com/JLSHuN



```java
class Superclass{

   int age;

   Superclass(int age){
      this.age=age;
   }

   public void getAge(){
      System.out.println("The value of the variable named age in super class is: " +age);
   }

}

class Subclass extends Superclass {

   Subclass(int age){
      super(age);
   }

   public static void main(String argd[]){
      Subclass s= new Subclass(24);
      s.getAge();
   }

}
```

stdout

```
The value of the variable named age in super class is: 24
```

- You know “this” in Java
- You also know “super” in Java

- So “this” course is getting “super” now!