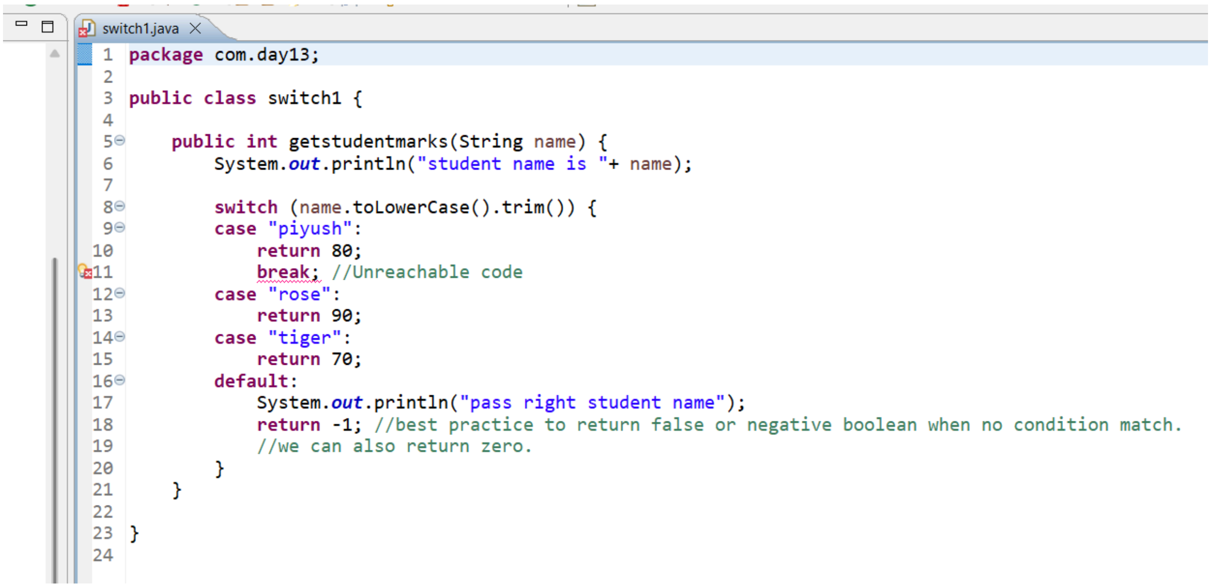
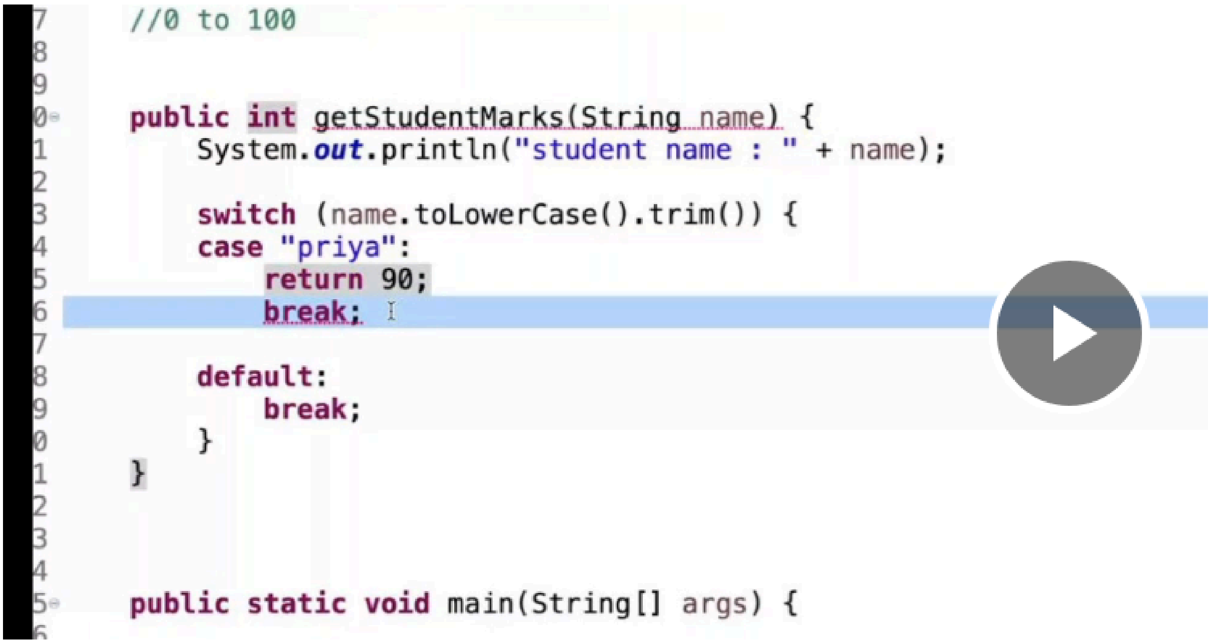
We get error when trying to add return and break together in switch case-

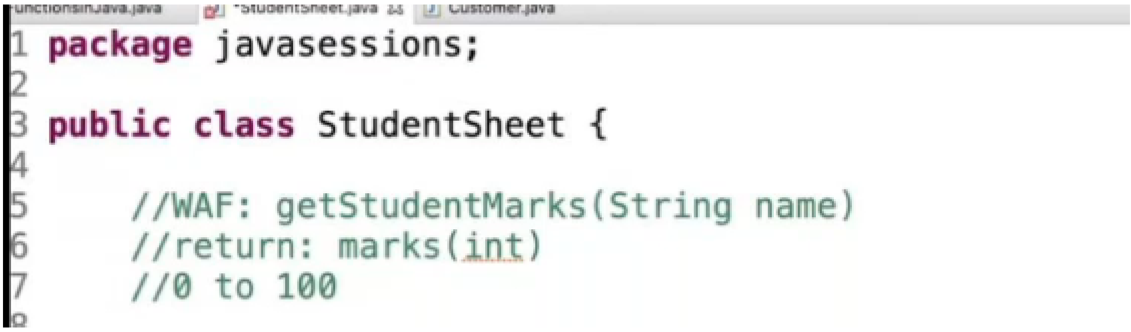
paste the switch1



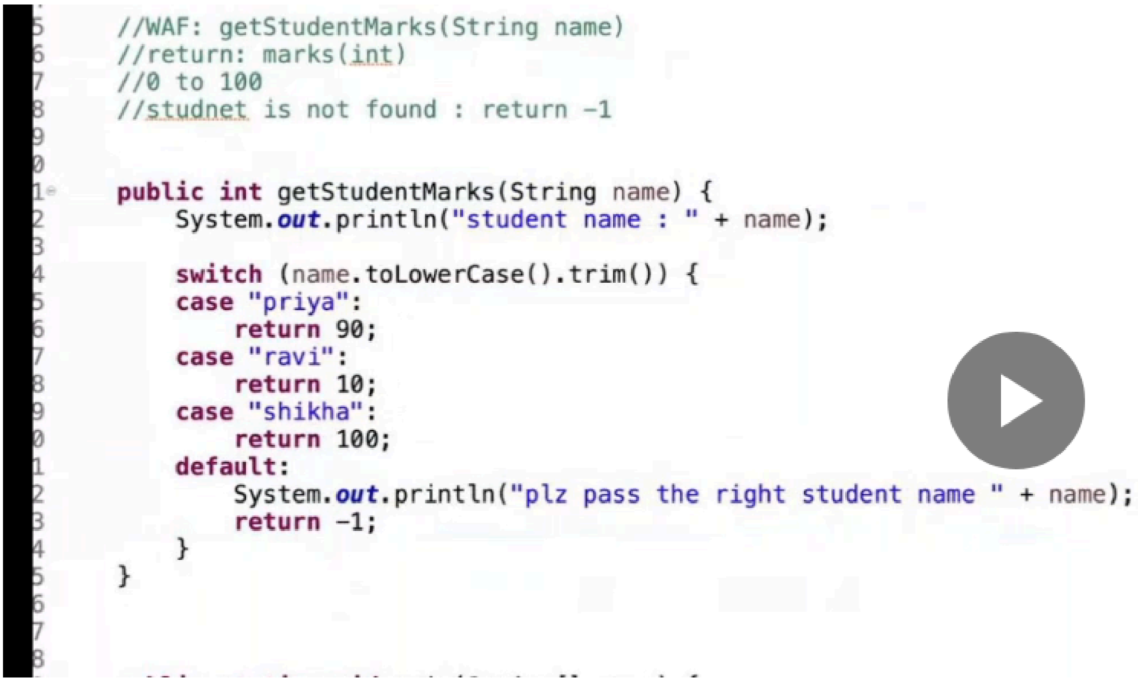
```java
package com.day13;

public class switch1 {

    public int getstudentmarks(String name) {
        System.out.println("student name is "+ name);

        switch (name.toLowerCase().trim()) {
        case "piyush":
            return 80;
            break; //Unreachable code
        case "rose":
            return 90;
        case "tiger":
            return 70;
        default:
            System.out.println("pass right student name");
            return -1; //best practice to return false or negative boolean when no condition match.
            //we can also return zero.
        }
    }

}
```



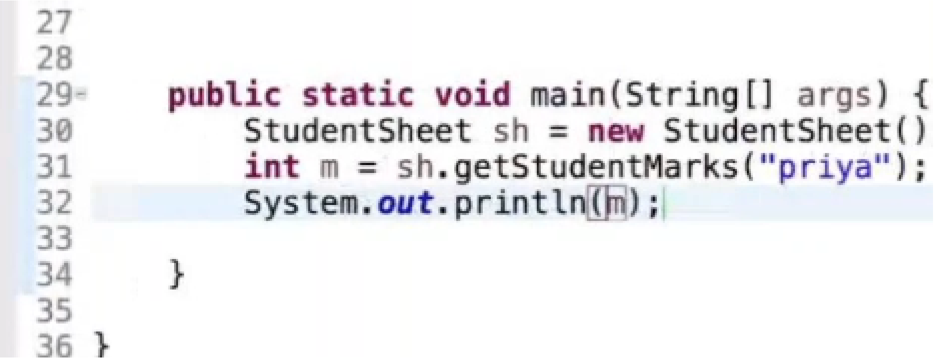
```java
    //0 to 100


    public int getStudentMarks(String name) {
        System.out.println("student name : " + name);

        switch (name.toLowerCase().trim()) {
        case "priya":
            return 90;
            break;

        default:
            break;
        }
    }




    public static void main(String[] args) {
```



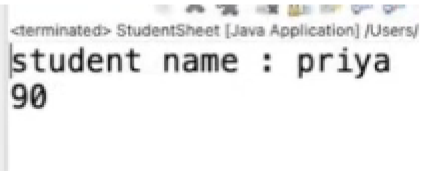
```java
package javasessions;

public class StudentSheet {

    //WAF: getStudentMarks(String name)
    //return: marks(int)
    //0 to 100
```



```java
    //WAF: getStudentMarks(String name)
    //return: marks(int)
    //0 to 100
    //studnet is not found : return -1


    public int getStudentMarks(String name) {
        System.out.println("student name : " + name);

        switch (name.toLowerCase().trim()) {
        case "priya":
            return 90;
        case "ravi":
            return 10;
        case "shikha":
            return 100;
        default:
            System.out.println("plz pass the right student name " + name);
            return -1;
        }
    }
```

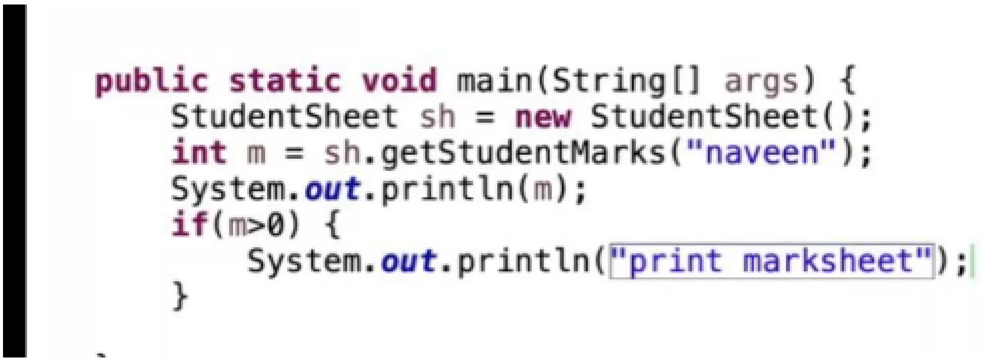


```java


    public static void main(String[] args) {
        StudentSheet sh = new StudentSheet()
        int m = sh.getStudentMarks("priya");
        System.out.println(m);

    }

}
```

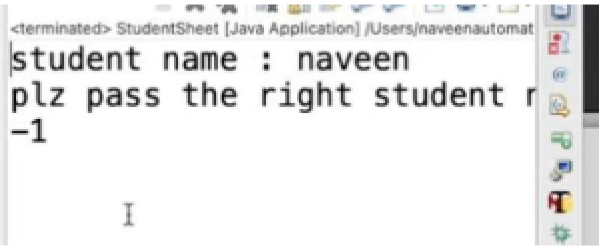


```
<terminated> StudentSheet [Java Application] /Users/
student name : priya
90
```

Use the values returned by method-



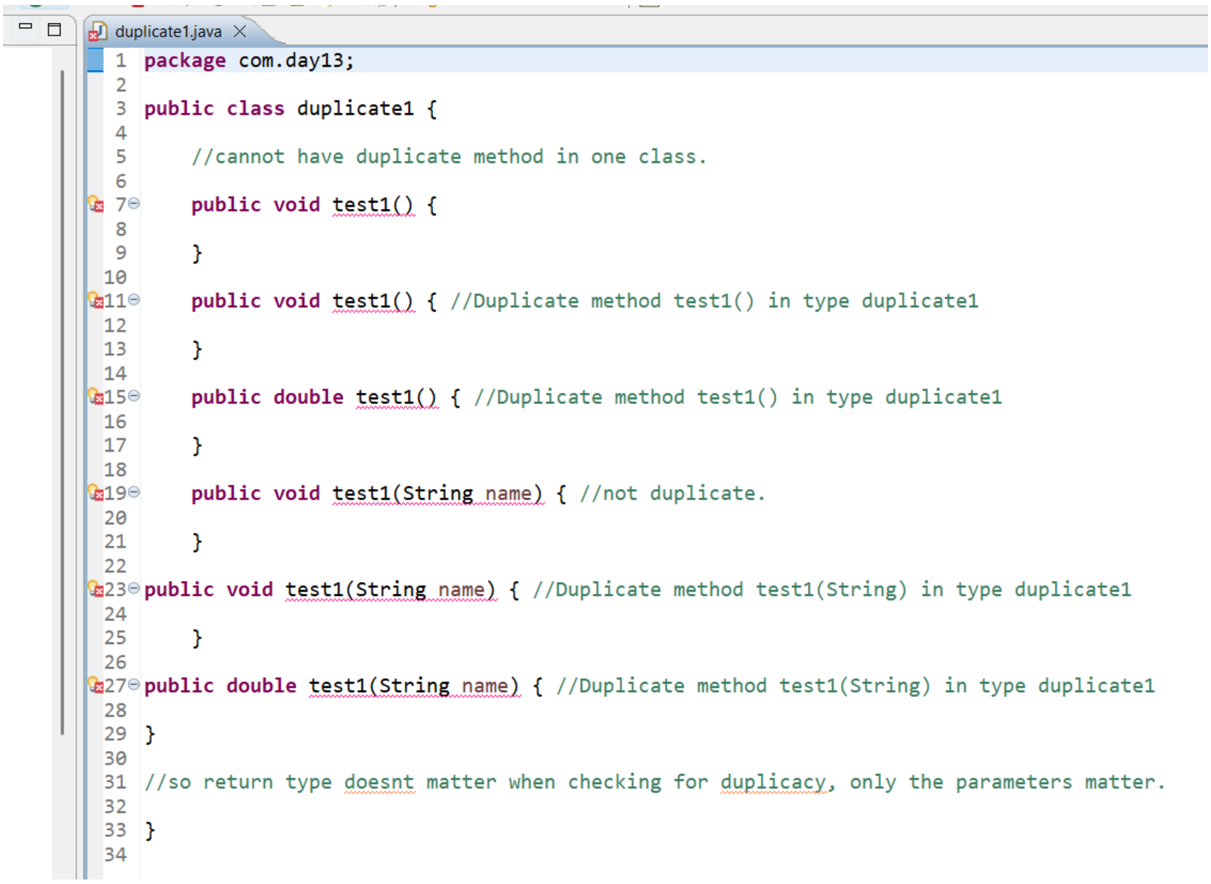
```
public static void main(String[] args) {
    StudentSheet sh = new StudentSheet();
    int m = sh.getStudentMarks("naveen");
    System.out.println(m);
    if(m>0) {
        System.out.println("print marksheet");
    }
```



```
<terminated> StudentSheet [Java Application] /Users/naveenautomat
student name : naveen
plz pass the right student r
-1
```

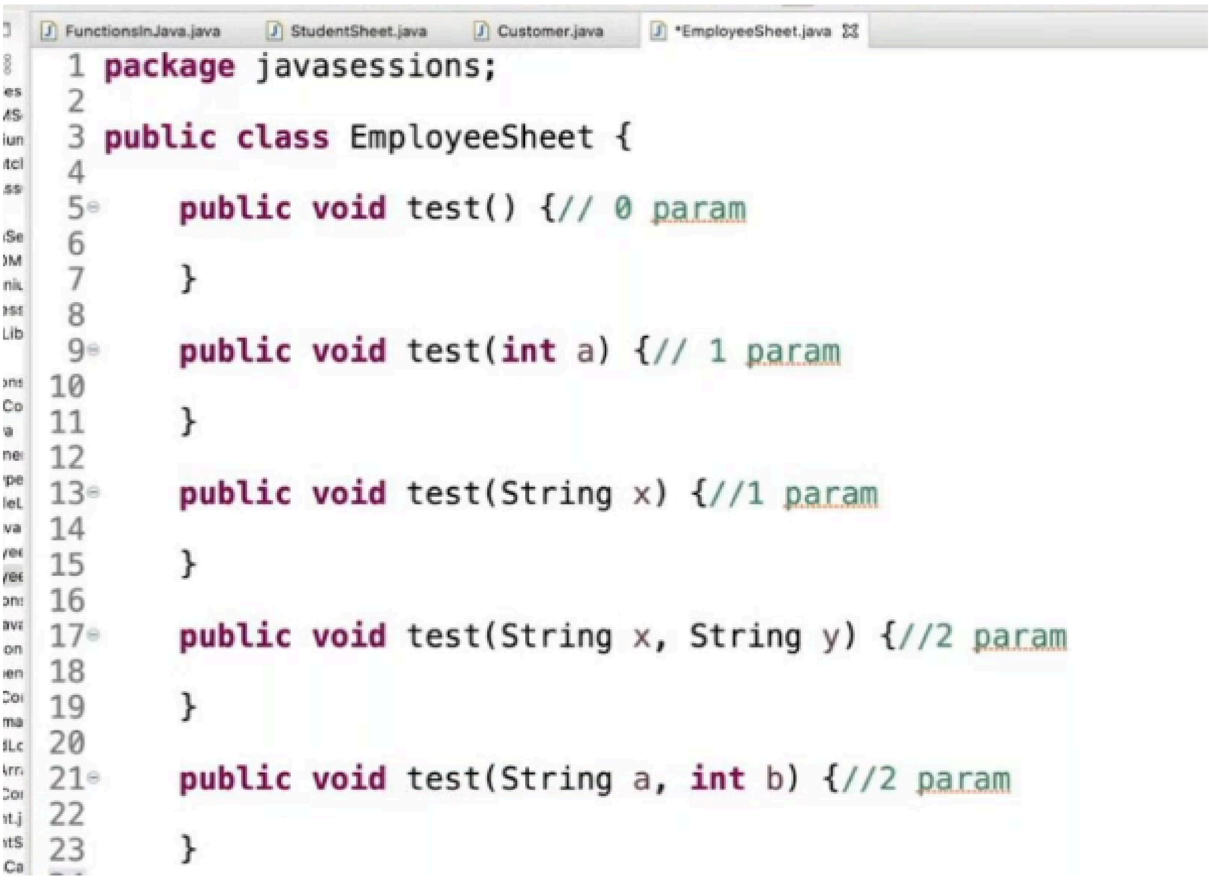
Duplicate methods in one class not allowed-

paste duplicate1

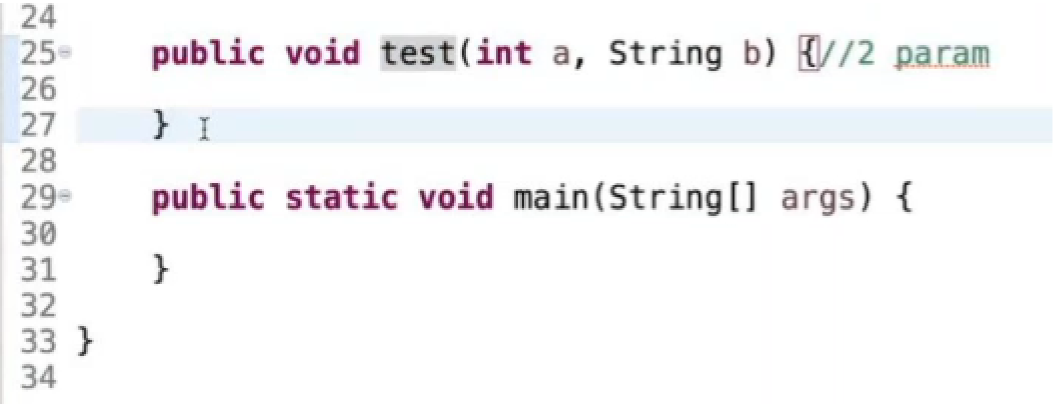


```
package com.day13;

public class duplicate1 {

    //cannot have duplicate method in one class.

    public void test1() {

    }

    public void test1() { //Duplicate method test1() in type duplicate1

    }

    public double test1() { //Duplicate method test1() in type duplicate1

    }

    public void test1(String name) { //not duplicate.

    }

public void test1(String name) { //Duplicate method test1(String) in type duplicate1

    }

public double test1(String name) { //Duplicate method test1(String) in type duplicate1

}

//so return type doesnt matter when checking for duplicacy, only the parameters matter.

}

```

Method overloading-



```
package javasessions;

public class EmployeeSheet {

    public void test() {// 0 param

    }

    public void test(int a) {// 1 param

    }

    public void test(String x) {//1 param

    }

    public void test(String x, String y) {//2 param

    }

    public void test(String a, int b) {//2 param

    }
```



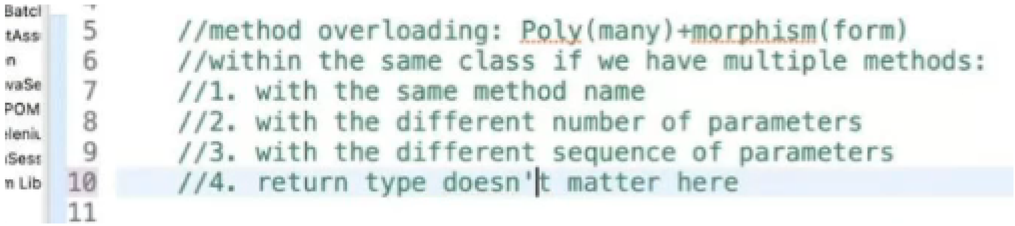
```

    public void test(int a, String b) {//2 param

    }

    public static void main(String[] args) {

    }

}

```

Also known as polymorphism.

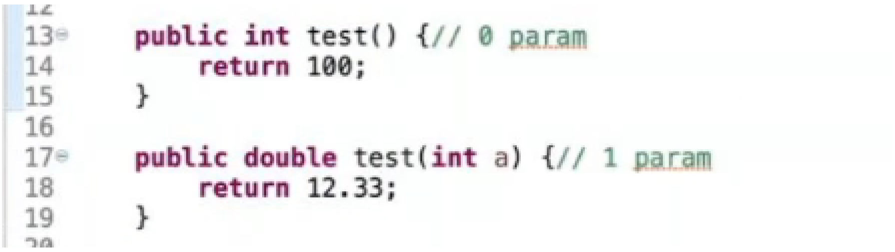
Poly means many.

Morphism means different forms.

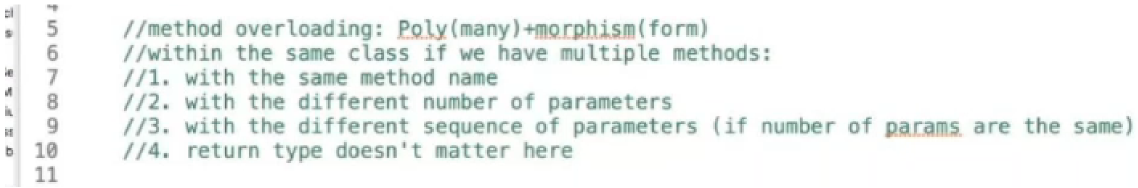
Within same class.



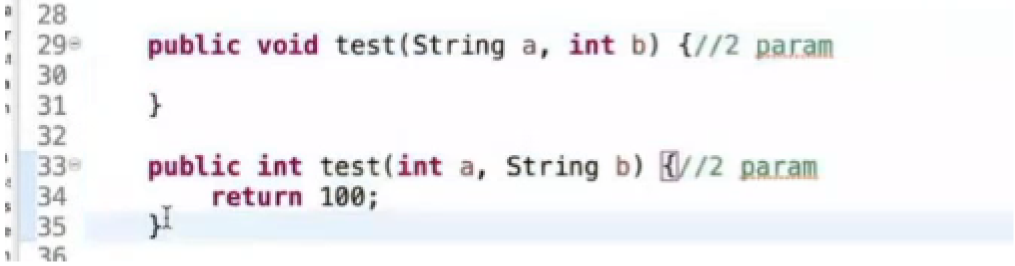
```
    //method overloading: Poly(many)+morphism(form)
    //within the same class if we have multiple methods:
    //1. with the same method name
    //2. with the different number of parameters
    //3. with the different sequence of parameters
    //4. return type doesn't matter here
```



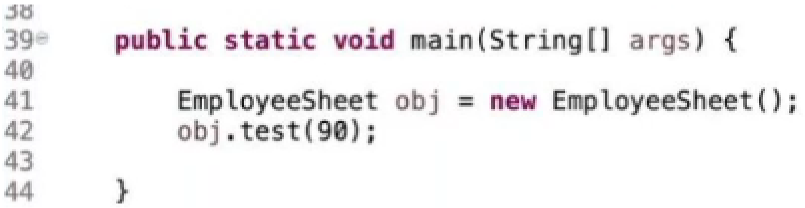
```
    public int test() {// 0 param
        return 100;
    }

    public double test(int a) {// 1 param
        return 12.33;
    }
```



```
    //method overloading: Poly(many)+morphism(form)
    //within the same class if we have multiple methods:
    //1. with the same method name
    //2. with the different number of parameters
    //3. with the different sequence of parameters (if number of params are the same)
    //4. return type doesn't matter here
```



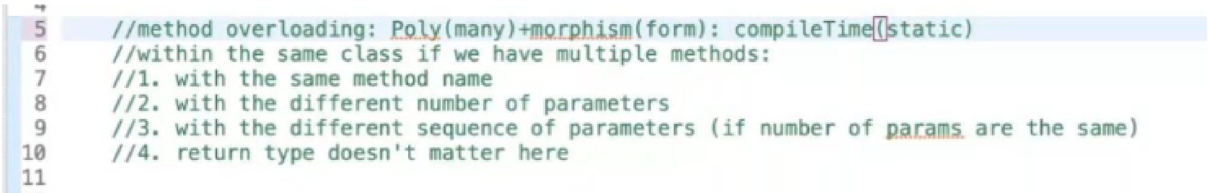
```
    public void test(String a, int b) {//2 param

    }

    public int test(int a, String b) {//2 param
        return 100;
    }
```



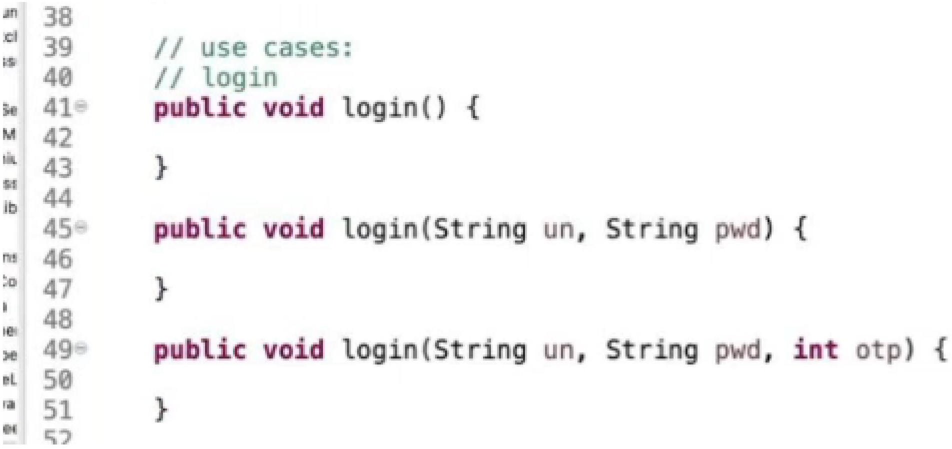
```
    public static void main(String[] args) {

        EmployeeSheet obj = new EmployeeSheet();
        obj.test(90);

    }
```

Test(int a) method called.

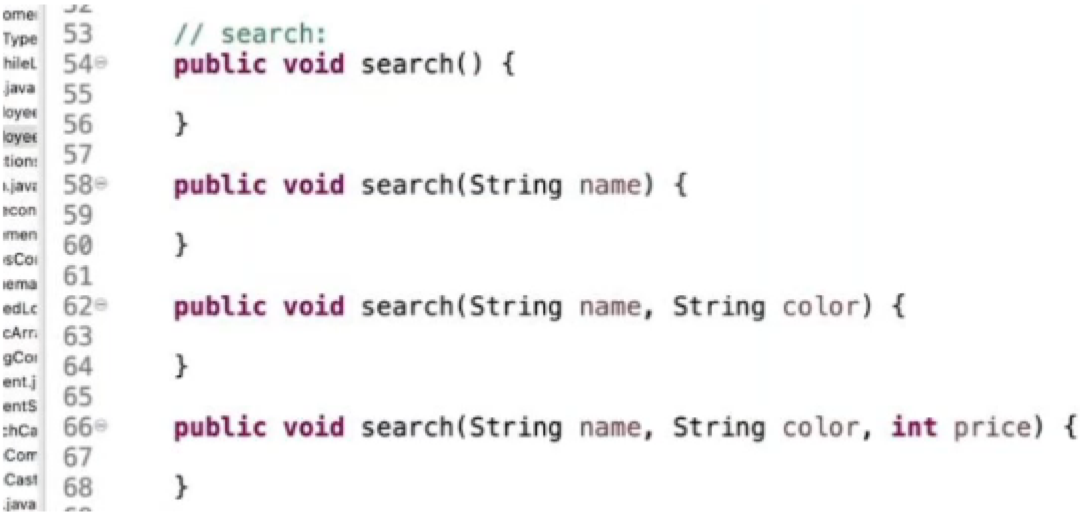
Compiler knows which method to call.



```
    //method overloading: Poly(many)+morphism(form): compileTime(static)
    //within the same class if we have multiple methods:
    //1. with the same method name
    //2. with the different number of parameters
    //3. with the different sequence of parameters (if number of params are the same)
    //4. return type doesn't matter here
```



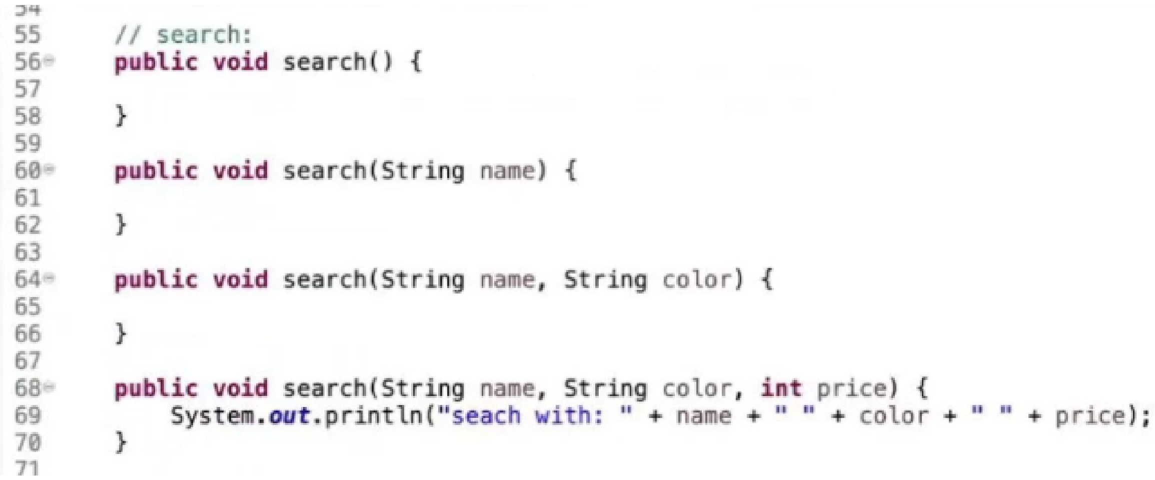
```java

    // use cases:
    // login
    public void login() {

    }

    public void login(String un, String pwd) {

    }

    public void login(String un, String pwd, int otp) {

    }
```



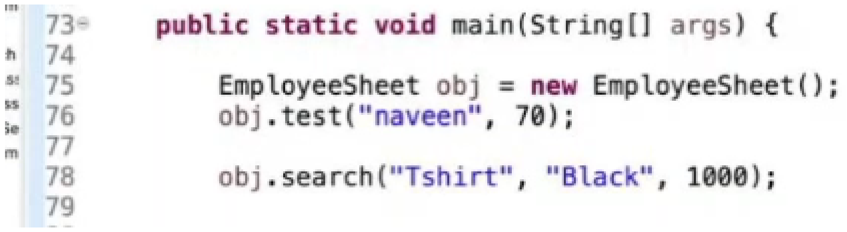
```java
    // search:
    public void search() {

    }

    public void search(String name) {

    }

    public void search(String name, String color) {

    }

    public void search(String name, String color, int price) {

    }
```

```java
    //will be used for same functionality (login, search, payment)
```

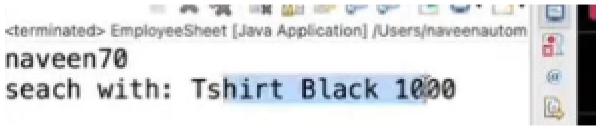
Maintenance easy due to modularity of the code.



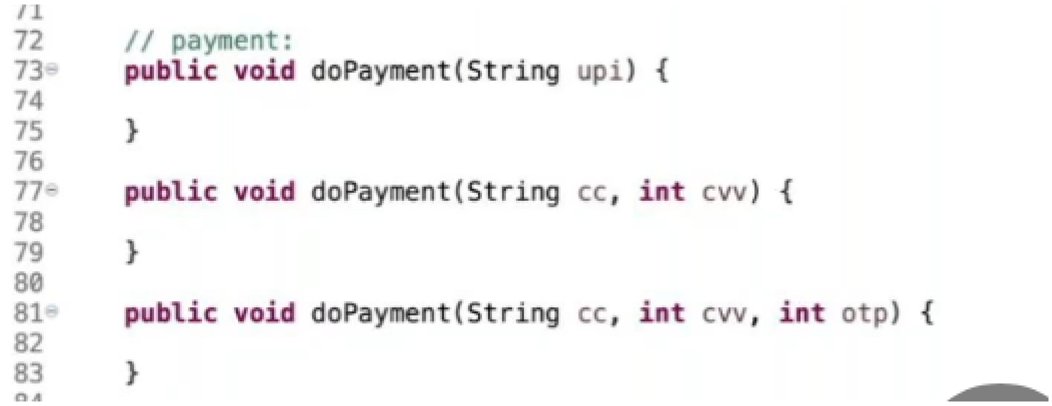
```
// search:
public void search() {

}

public void search(String name) {

}

public void search(String name, String color) {

}

public void search(String name, String color, int price) {
    System.out.println("seach with: " + name + " " + color + " " + price);
}
```



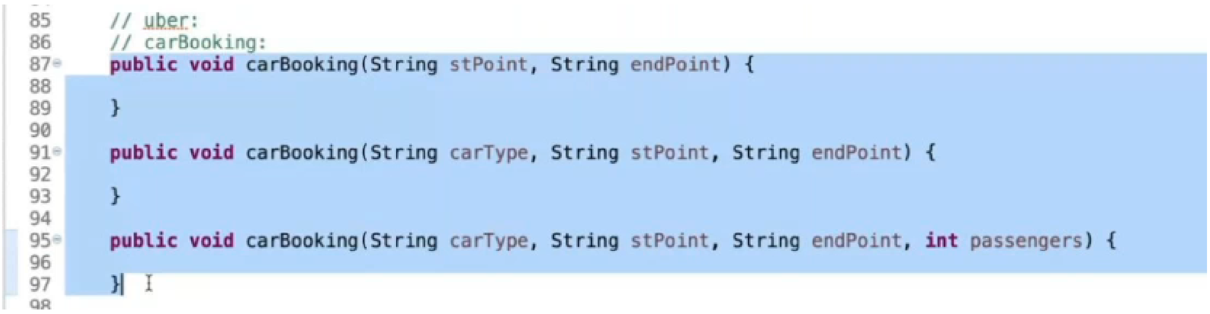
```
public static void main(String[] args) {

    EmployeeSheet obj = new EmployeeSheet();
    obj.test("naveen", 70);

    obj.search("Tshirt", "Black", 1000);

```



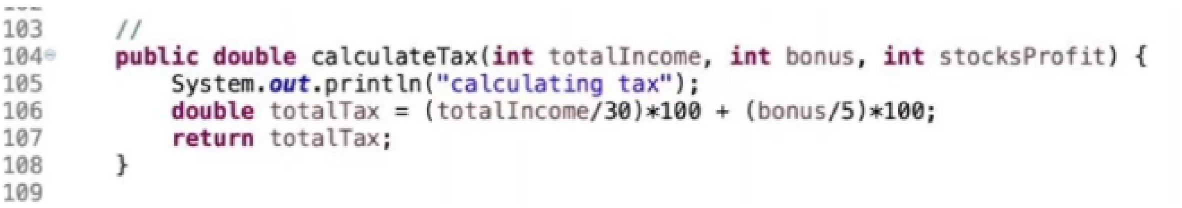
```
<terminated> EmployeeSheet [Java Application] /Users/naveenautom
naveen70
seach with: Tshirt Black 1000
```



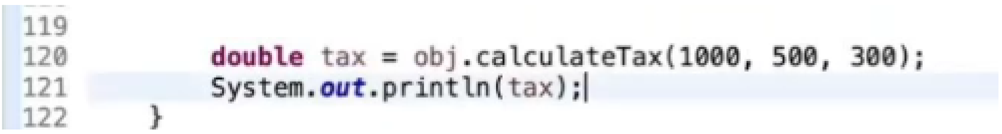
```
// payment:
public void doPayment(String upi) {

}

public void doPayment(String cc, int cvv) {

}

public void doPayment(String cc, int cvv, int otp) {

}
```



```java
// uber:
// carBooking:
public void carBooking(String stPoint, String endPoint) {

}

public void carBooking(String carType, String stPoint, String endPoint) {

}

public void carBooking(String carType, String stPoint, String endPoint, int passengers) {

}
```

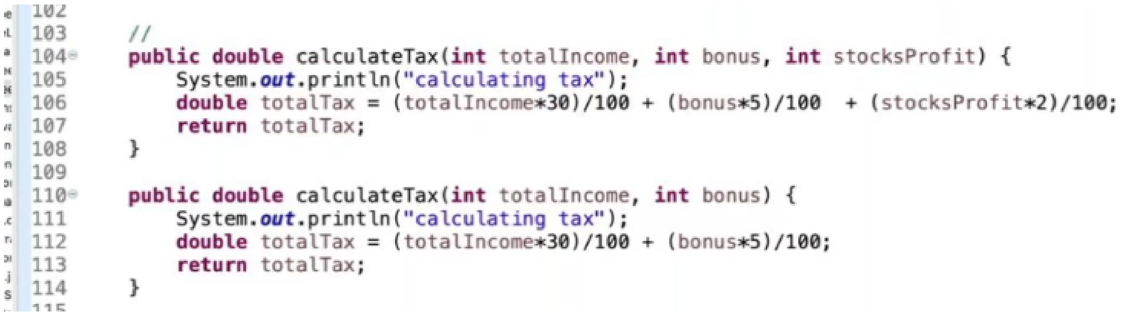


```java
//
public double calculateTax(int totalIncome, int bonus, int stocksProfit) {
    System.out.println("calculating tax");
    double totalTax = (totalIncome/30)*100 + (bonus/5)*100;
    return totalTax;
}
```

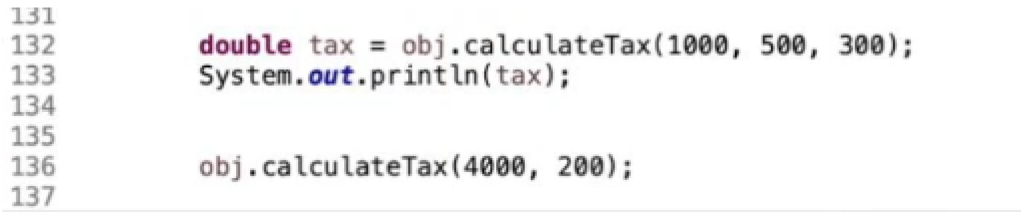


```java
        double tax = obj.calculateTax(1000, 500, 300);
        System.out.println(tax);
    }
```

13300.00.



```java
//
public double calculateTax(int totalIncome, int bonus, int stocksProfit) {
    System.out.println("calculating tax");
    double totalTax = (totalIncome*30)/100 + (bonus*5)/100  + (stocksProfit*2)/100;
    return totalTax;
}

public double calculateTax(int totalIncome, int bonus) {
    System.out.println("calculating tax");
    double totalTax = (totalIncome*30)/100 + (bonus*5)/100;
    return totalTax;
}
```



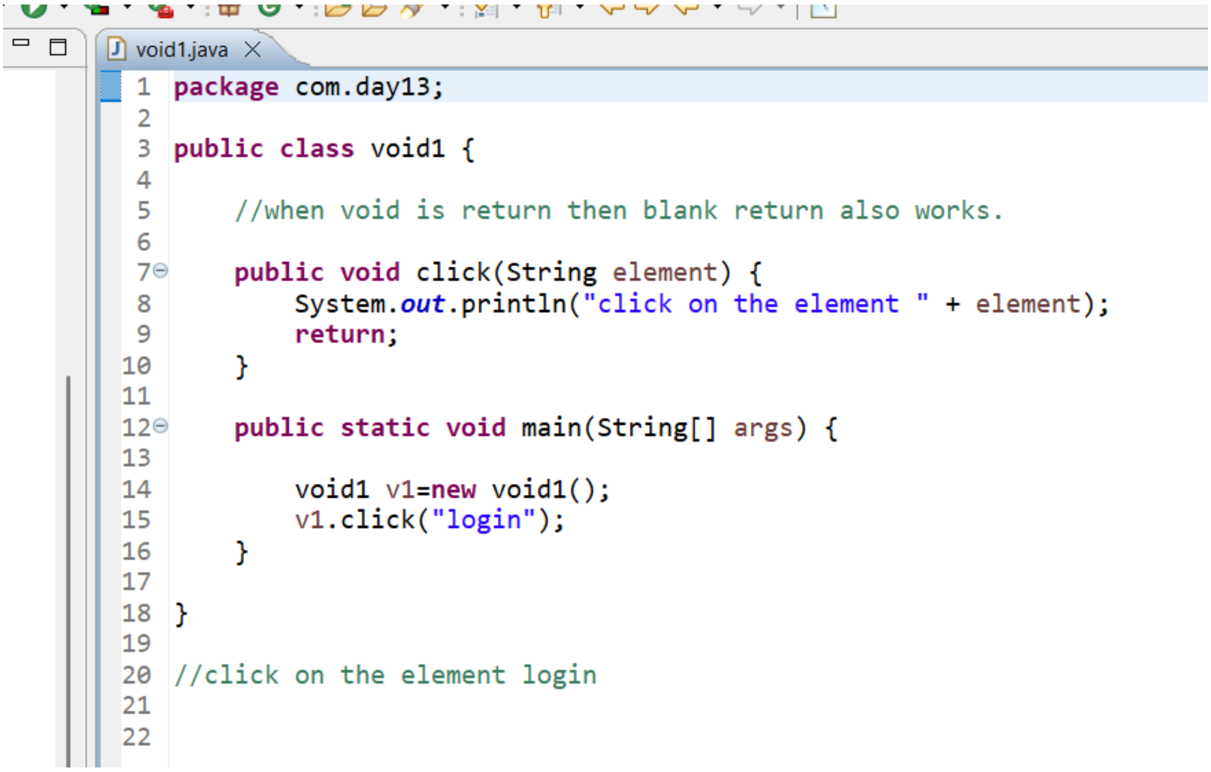
```java
double tax = obj.calculateTax(1000, 500, 300);
System.out.println(tax);


obj.calculateTax(4000, 200);
```

Note-

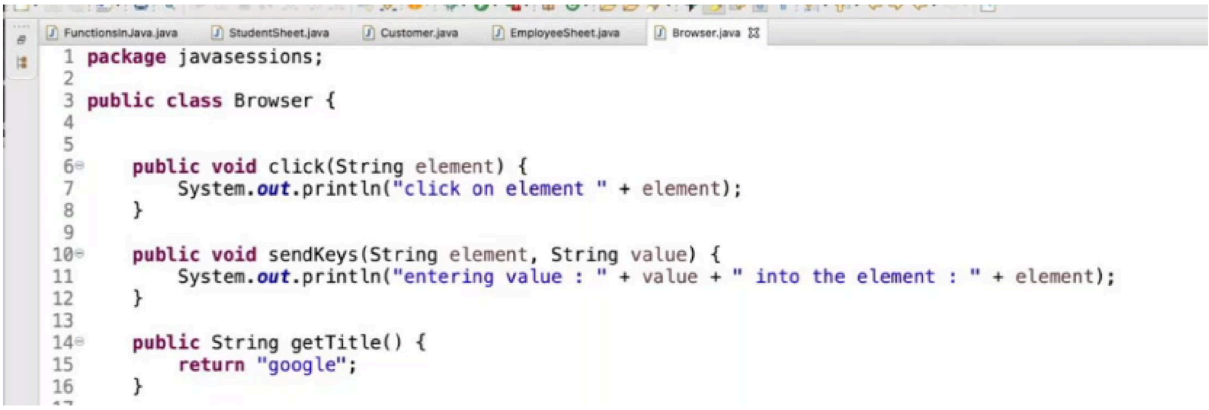
All the parameters should be used within the body of method, else not good practice.

Void return-

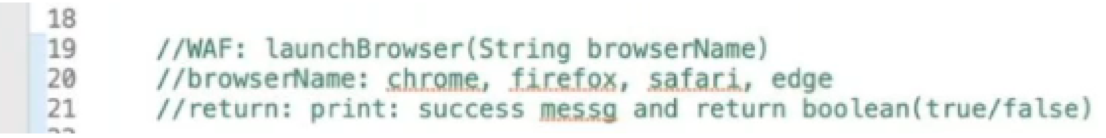
paste void1



```java
package com.day13;

public class void1 {

    //when void is return then blank return also works.

    public void click(String element) {
        System.out.println("click on the element " + element);
        return;
    }

    public static void main(String[] args) {

        void1 v1=new void1();
        v1.click("login");
    }

}

//click on the element login

```



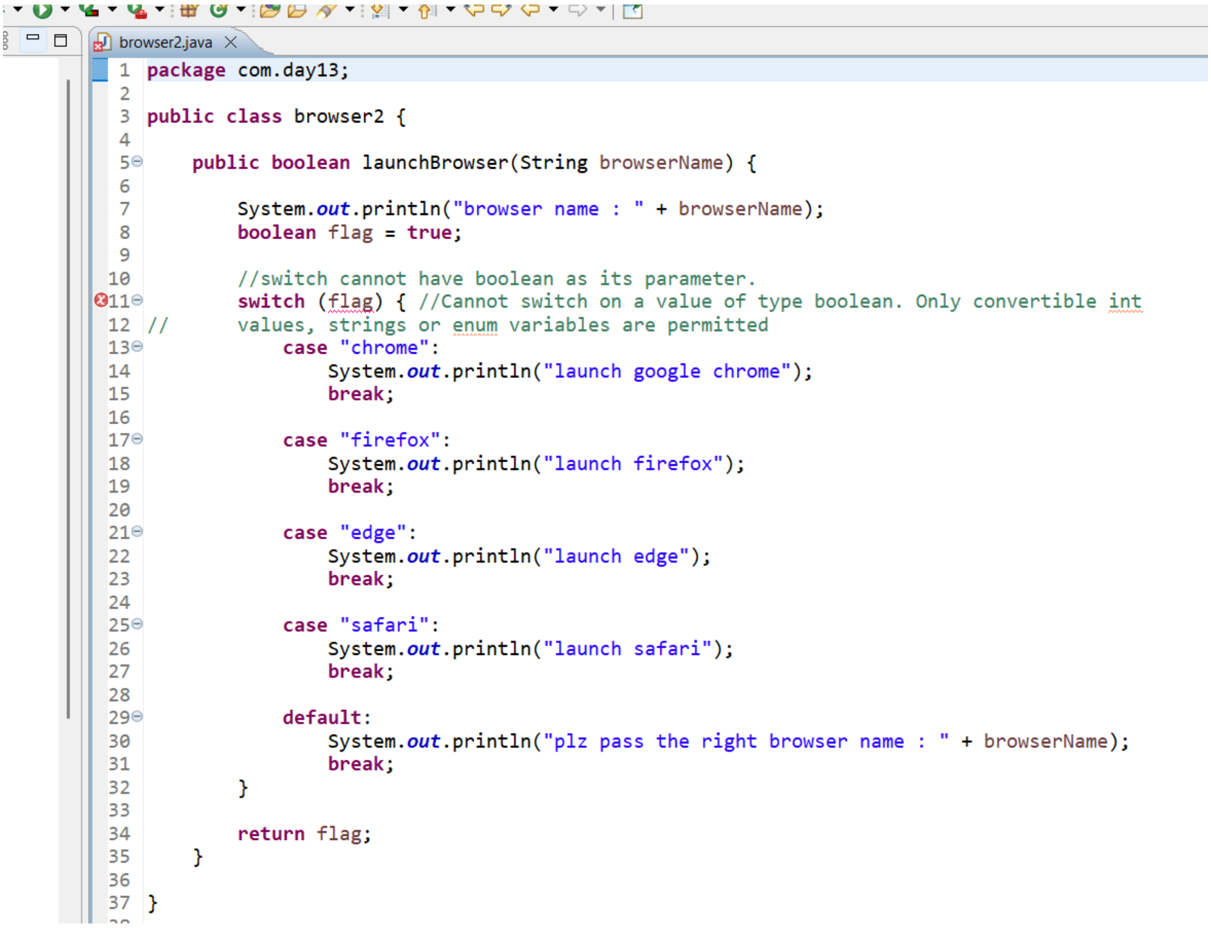
```java
package javasessions;

public class Browser {


    public void click(String element) {
        System.out.println("click on element " + element);
    }

    public void sendKeys(String element, String value) {
        System.out.println("entering value : " + value + " into the element : " + element);
    }

    public String getTitle() {
        return "google";
    }
```



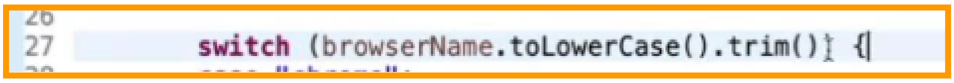
```java

    //WAF: launchBrowser(String browserName)
    //browserName: chrome, firefox, safari, edge
    //return: print: success messg and return boolean(true/false)
```

Boolean not allowed in switch() –

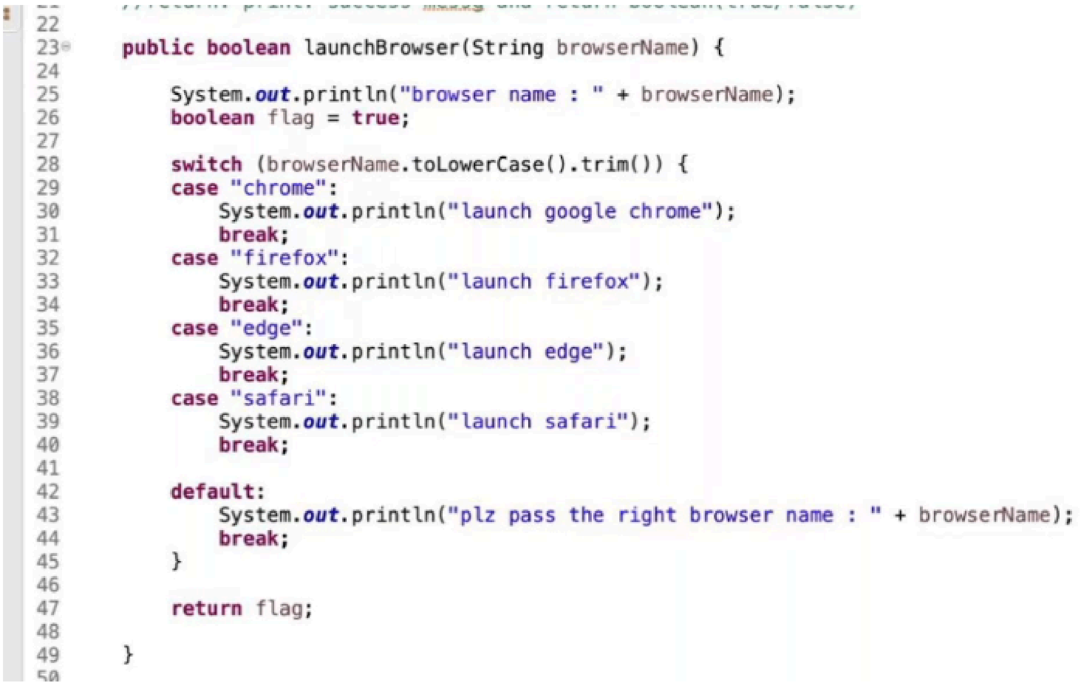
paste browser2-



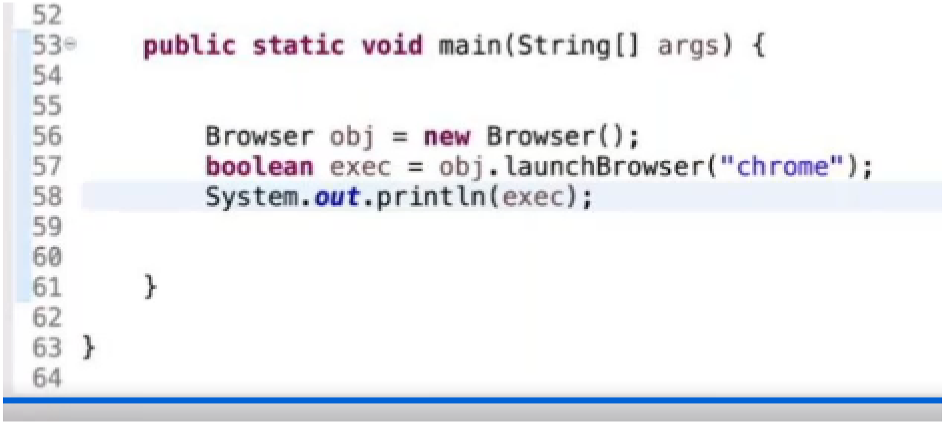
```java
package com.day13;

public class browser2 {

    public boolean launchBrowser(String browserName) {

        System.out.println("browser name : " + browserName);
        boolean flag = true;

        //switch cannot have boolean as its parameter.
        switch (flag) { //Cannot switch on a value of type boolean. Only convertible int
//      values, strings or enum variables are permitted
            case "chrome":
                System.out.println("launch google chrome");
                break;

            case "firefox":
                System.out.println("launch firefox");
                break;

            case "edge":
                System.out.println("launch edge");
                break;

            case "safari":
                System.out.println("launch safari");
                break;

            default:
                System.out.println("plz pass the right browser name : " + browserName);
                break;
        }

        return flag;
    }

}
```



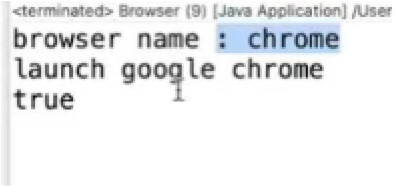
```java
        switch (browserName.toLowerCase().trim()) {
```



```
    //return: print: success msg and return boolean(true/false)

    public boolean launchBrowser(String browserName) {

        System.out.println("browser name : " + browserName);
        boolean flag = true;

        switch (browserName.toLowerCase().trim()) {
        case "chrome":
            System.out.println("launch google chrome");
            break;
        case "firefox":
            System.out.println("launch firefox");
            break;
        case "edge":
            System.out.println("launch edge");
            break;
        case "safari":
            System.out.println("launch safari");
            break;

        default:
            System.out.println("plz pass the right browser name : " + browserName);
            break;
        }

        return flag;

    }
```

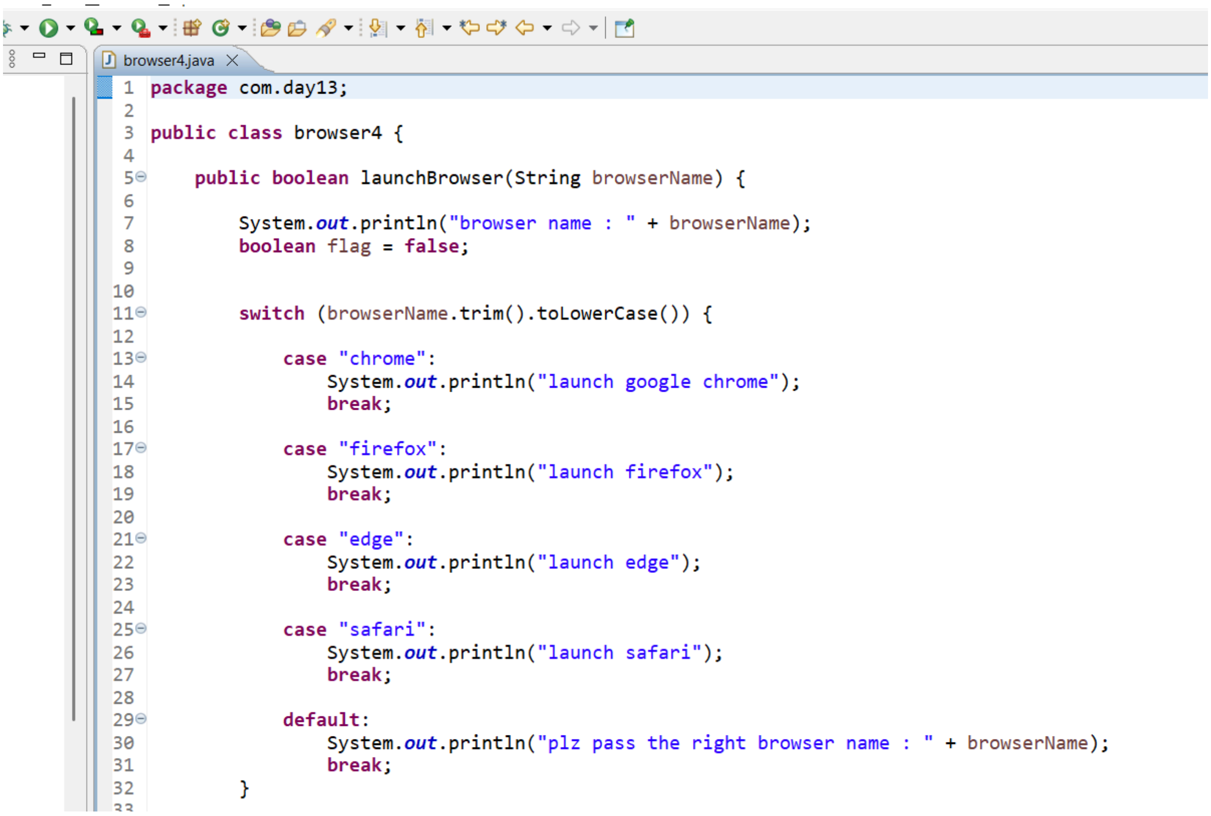


```
    public static void main(String[] args) {


        Browser obj = new Browser();
        boolean exec = obj.launchBrowser("chrome");
        System.out.println(exec);


    }

}
```

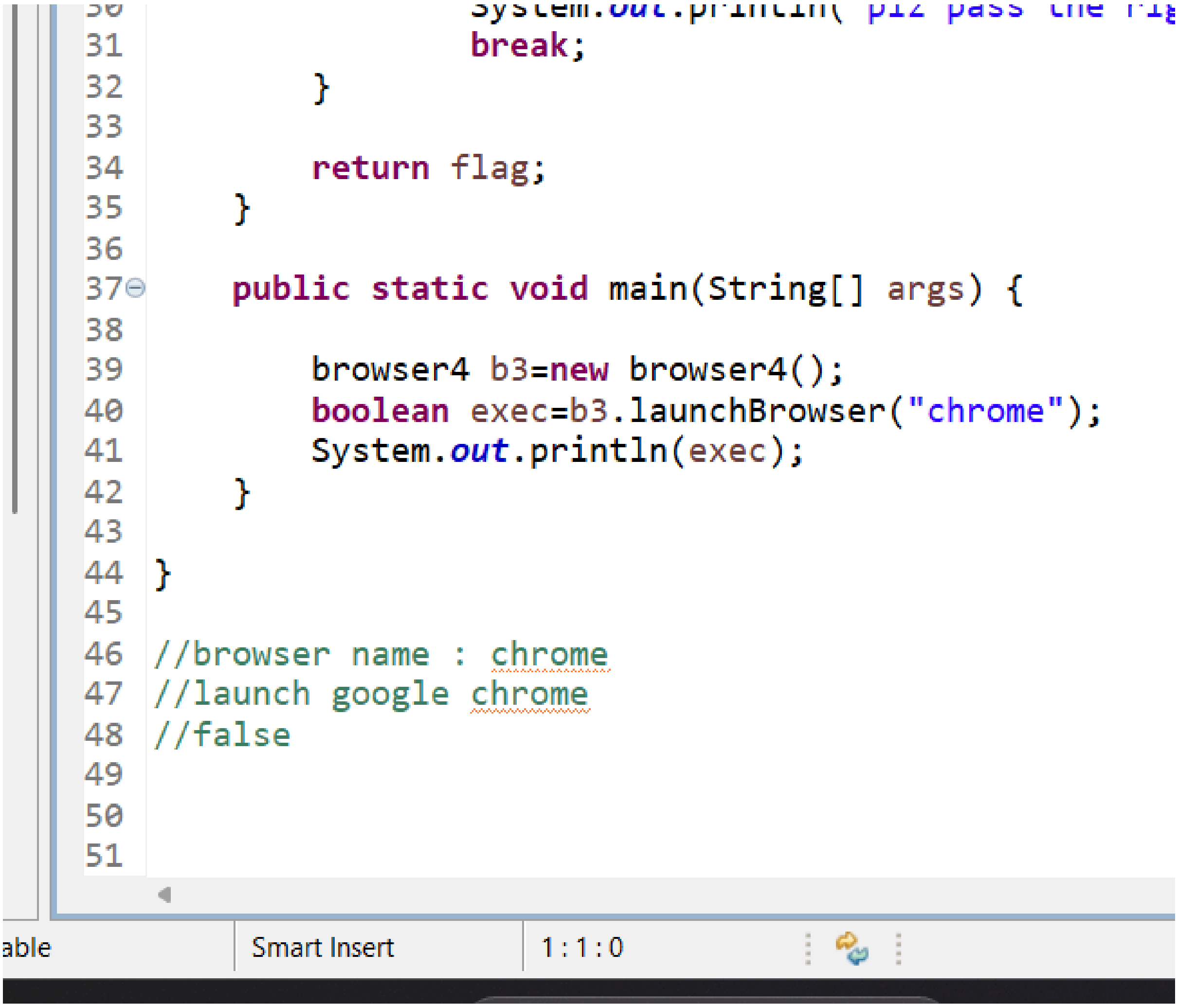


```
browser name : chrome
launch google chrome
true
```

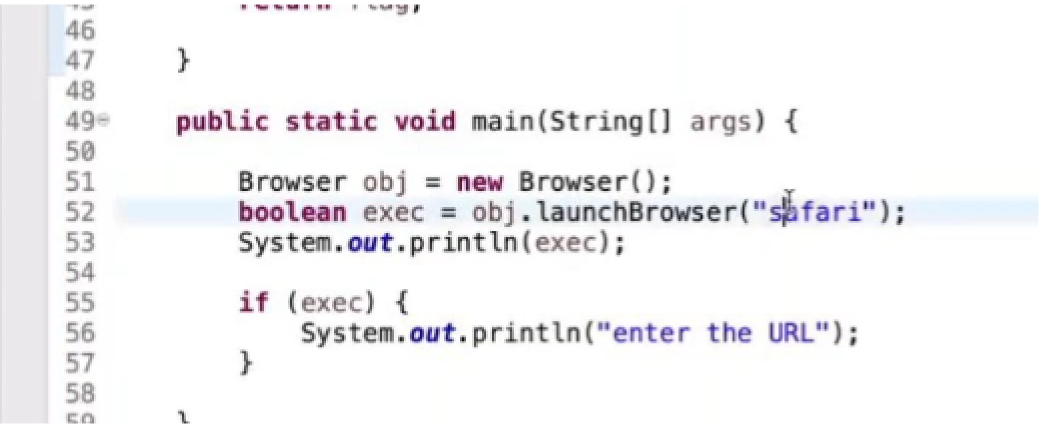
paste browser4



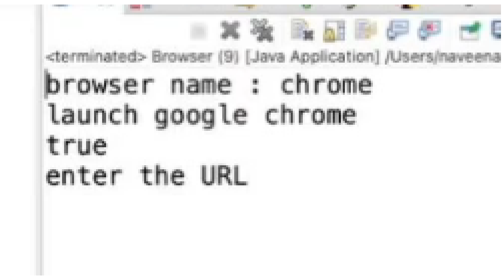
```
package com.day13;

public class browser4 {

    public boolean launchBrowser(String browserName) {

        System.out.println("browser name : " + browserName);
        boolean flag = false;


        switch (browserName.trim().toLowerCase()) {

            case "chrome":
                System.out.println("launch google chrome");
                break;

            case "firefox":
                System.out.println("launch firefox");
                break;

            case "edge":
                System.out.println("launch edge");
                break;

            case "safari":
                System.out.println("launch safari");
                break;

            default:
                System.out.println("plz pass the right browser name : " + browserName);
                break;
        }
```



```
        System.out.println("plz pass the rig
        break;
    }

    return flag;
  }

  public static void main(String[] args) {

    browser4 b3=new browser4();
    boolean exec=b3.launchBrowser("chrome");
    System.out.println(exec);
  }

}

//browser name : chrome
//launch google chrome
//false
```



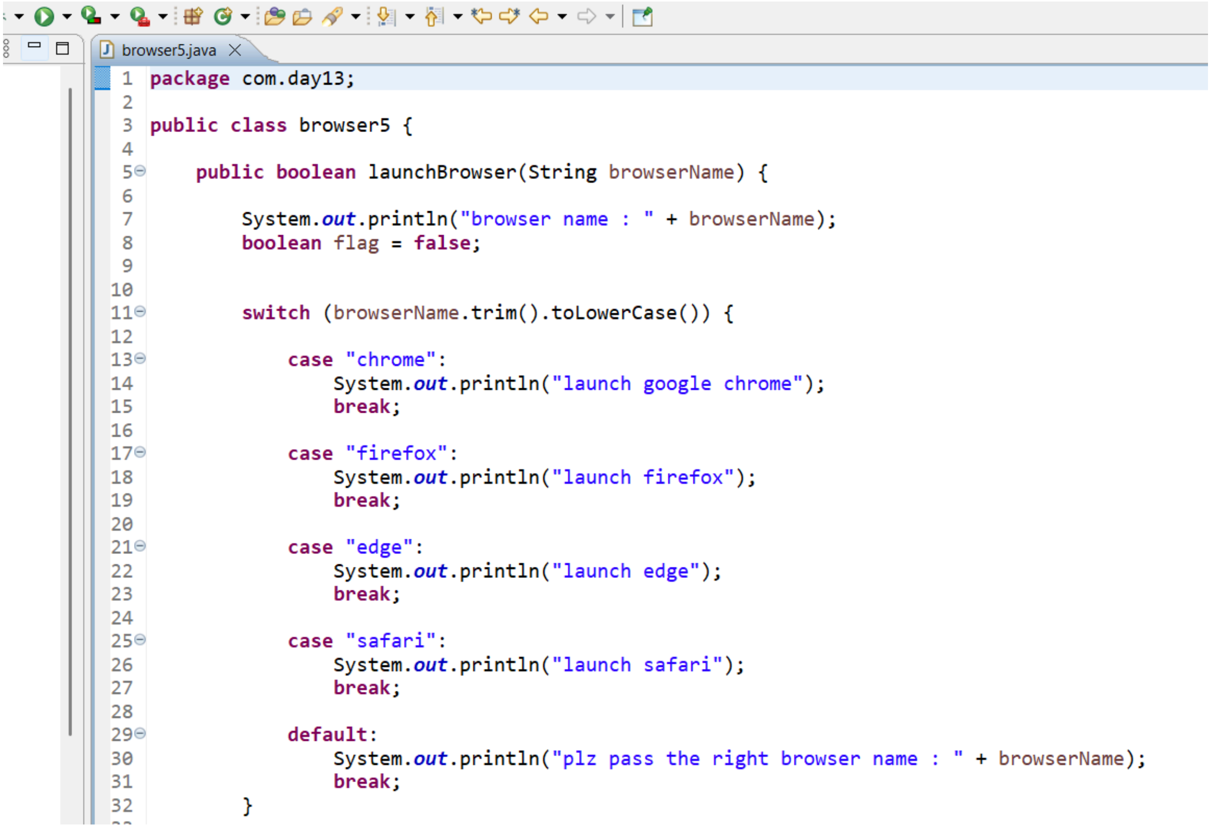
```
    return flag;
  }

  public static void main(String[] args) {

    Browser obj = new Browser();
    boolean exec = obj.launchBrowser("safari");
    System.out.println(exec);

    if (exec) {
      System.out.println("enter the URL");
    }

  }
```



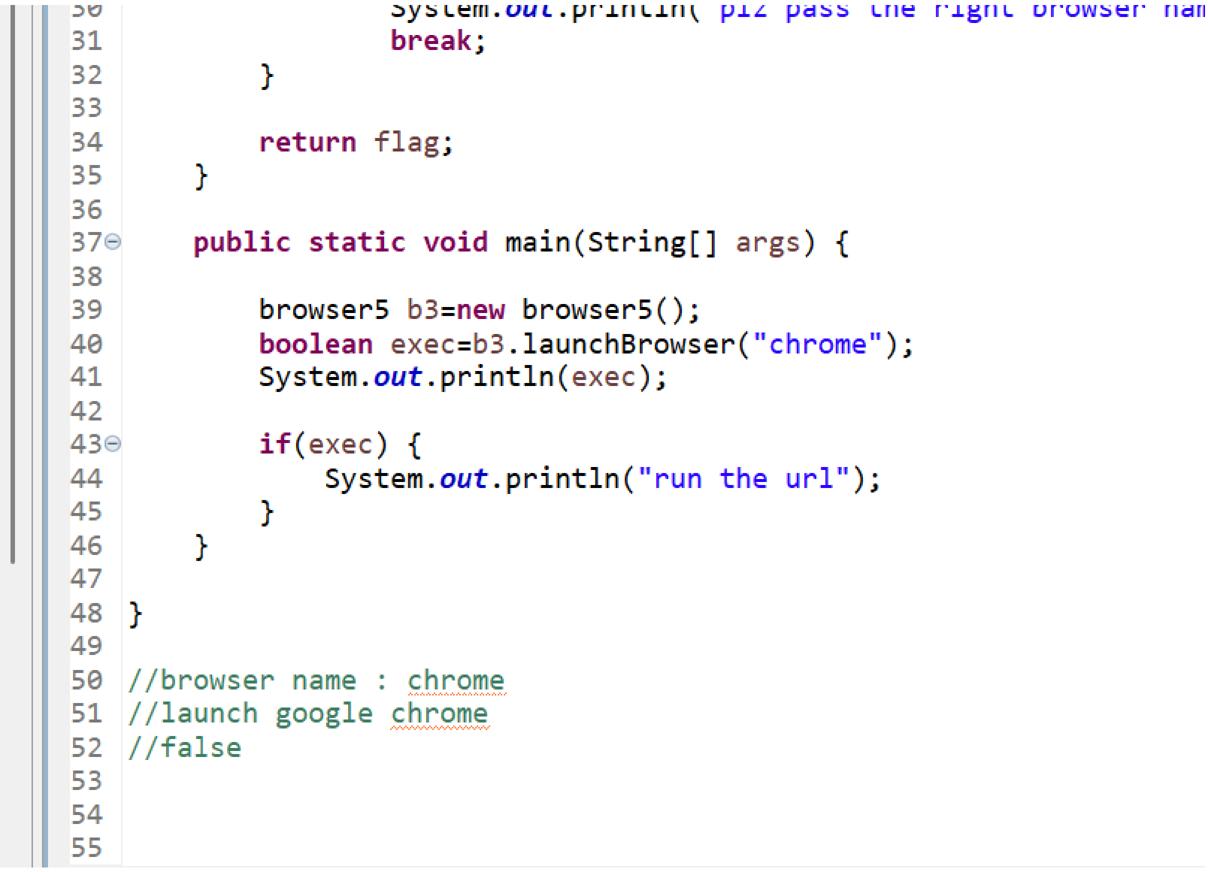
<terminated> Browser (9) [Java Application] /Users/naveena

```
browser name : chrome
launch google chrome
true
enter the URL
```

paste browser5



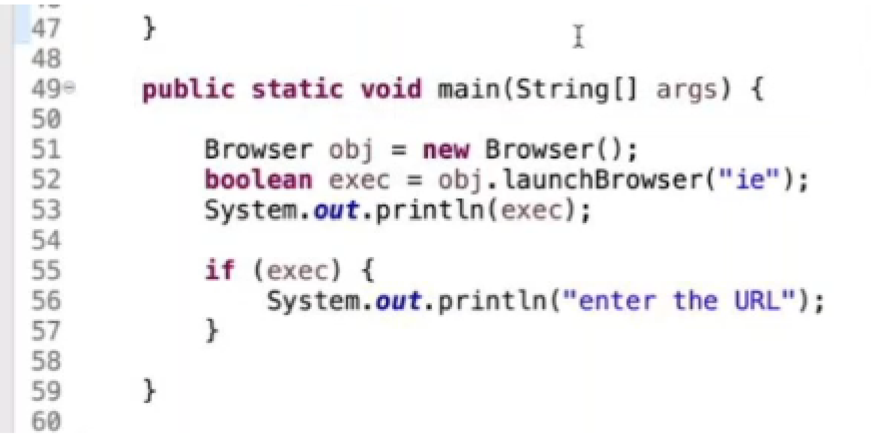
```java
package com.day13;

public class browser5 {

    public boolean launchBrowser(String browserName) {

        System.out.println("browser name : " + browserName);
        boolean flag = false;


        switch (browserName.trim().toLowerCase()) {

            case "chrome":
                System.out.println("launch google chrome");
                break;

            case "firefox":
                System.out.println("launch firefox");
                break;

            case "edge":
                System.out.println("launch edge");
                break;

            case "safari":
                System.out.println("launch safari");
                break;

            default:
                System.out.println("plz pass the right browser name : " + browserName);
                break;
        }
```



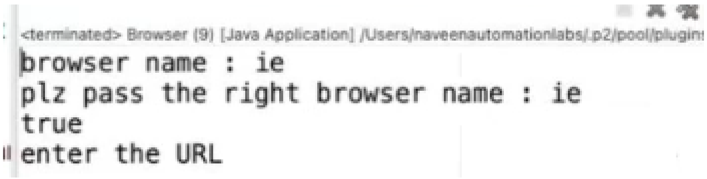
```
                System.out.println("plz pass the right browser nam
                break;
        }

        return flag;
    }

    public static void main(String[] args) {

        browser5 b3=new browser5();
        boolean exec=b3.launchBrowser("chrome");
        System.out.println(exec);

        if(exec) {
            System.out.println("run the url");
        }
    }

}

//browser name : chrome
//launch google chrome
//false
```

Wrong handling of flag-

Even with invalid browser, it gives flag true.

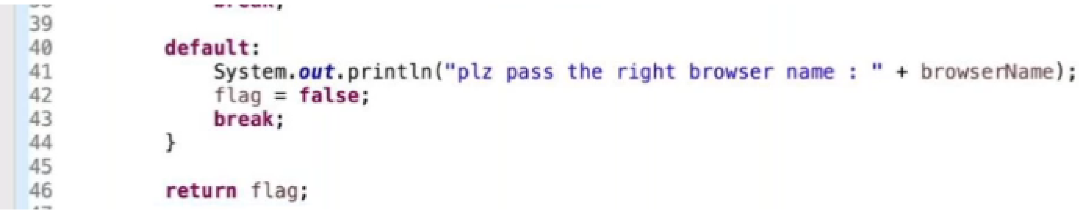


```
    }

    public static void main(String[] args) {

        Browser obj = new Browser();
        boolean exec = obj.launchBrowser("ie");
        System.out.println(exec);

        if (exec) {
            System.out.println("enter the URL");
        }

    }
```

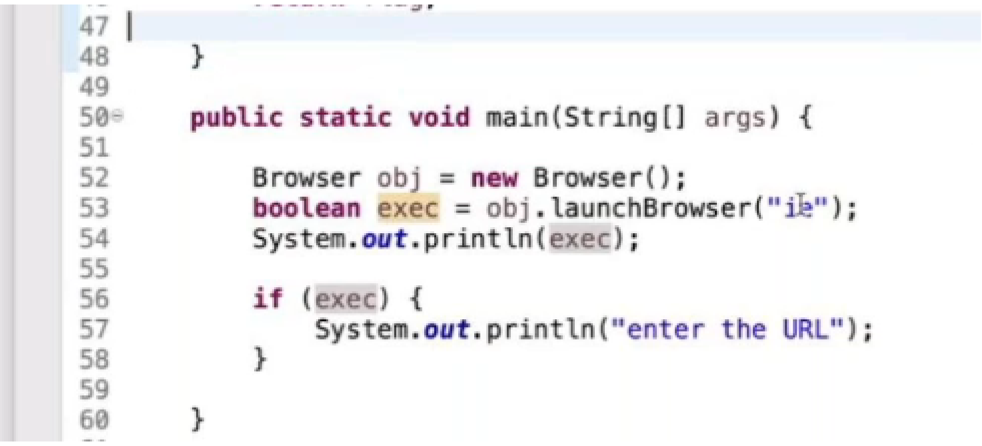


```
<terminated> Browser (9) [Java Application] /Users/naveenautomationlabs/.p2/pool/plugins
browser name : ie
plz pass the right browser name : ie
true
enter the URL
```

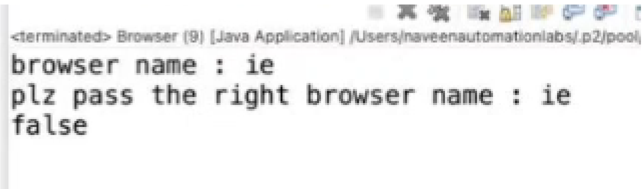
Proper way to use flags for invalid case-



```java
        default:
            System.out.println("plz pass the right browser name : " + browserName);
            flag = false;
            break;
        }

        return flag;
```



```java

    }

    public static void main(String[] args) {

        Browser obj = new Browser();
        boolean exec = obj.launchBrowser("ie");
        System.out.println(exec);

        if (exec) {
            System.out.println("enter the URL");
        }

    }
```

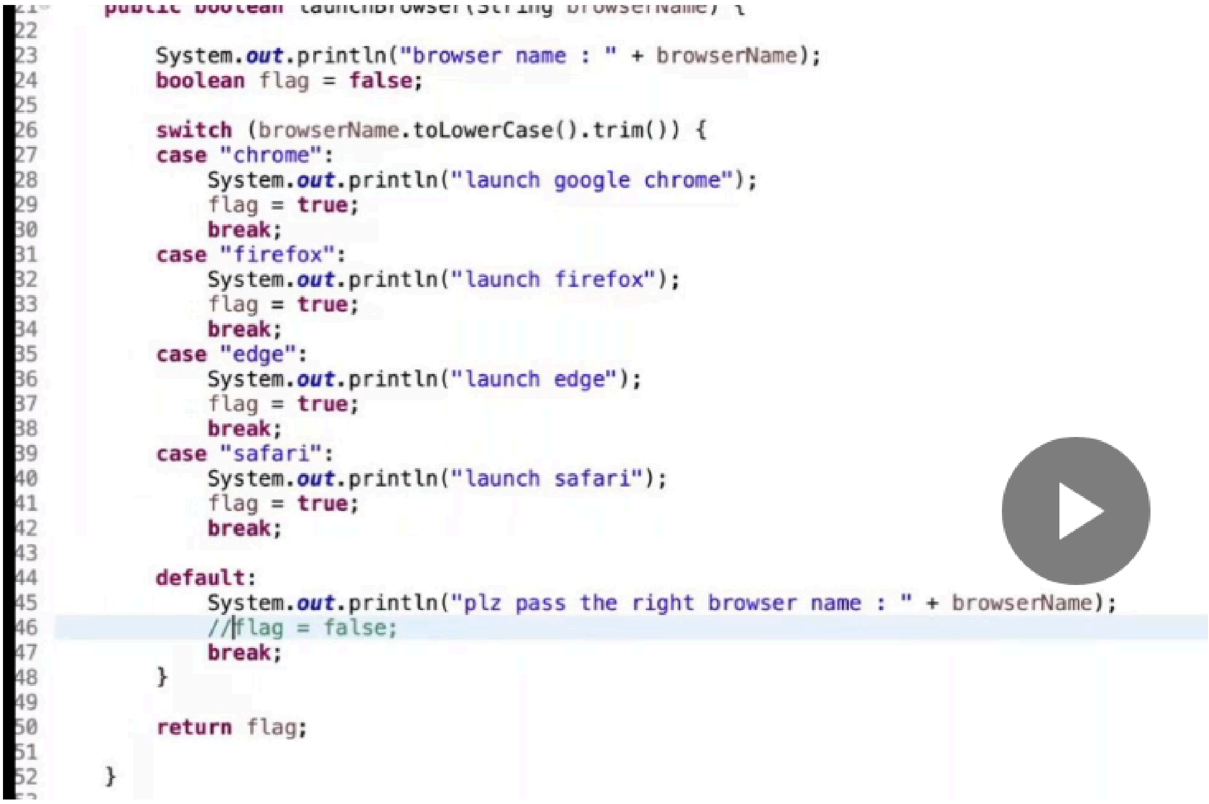


```
<terminated> Browser (9) [Java Application] /Users/naveenautomationlabs/.p2/pool/
browser name : ie
plz pass the right browser name : ie
false
```

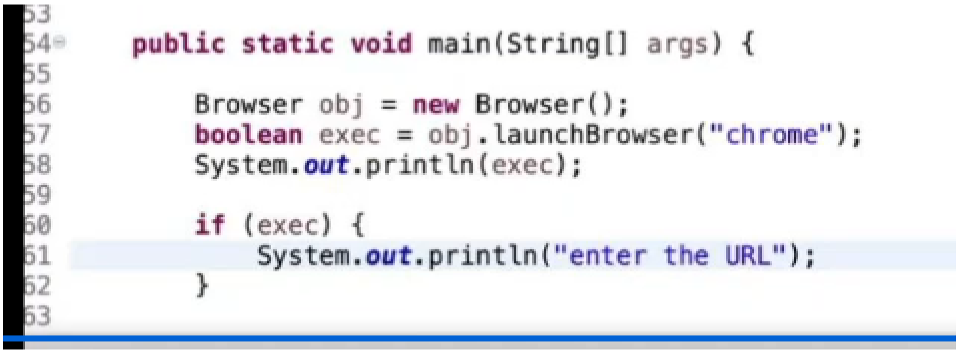
Start with negative value for flag-

//another way to handle flags in nice way.

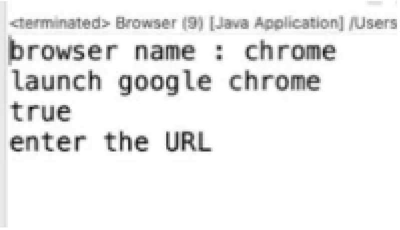
//start with false and make try and break when condition satisfied.



```java
    public boolean launchBrowser(String browserName) {

        System.out.println("browser name : " + browserName);
        boolean flag = false;

        switch (browserName.toLowerCase().trim()) {
        case "chrome":
            System.out.println("launch google chrome");
            flag = true;
            break;
        case "firefox":
            System.out.println("launch firefox");
            flag = true;
            break;
        case "edge":
            System.out.println("launch edge");
            flag = true;
            break;
        case "safari":
            System.out.println("launch safari");
            flag = true;
            break;

        default:
            System.out.println("plz pass the right browser name : " + browserName);
            //flag = false;
            break;
        }

        return flag;

    }
```



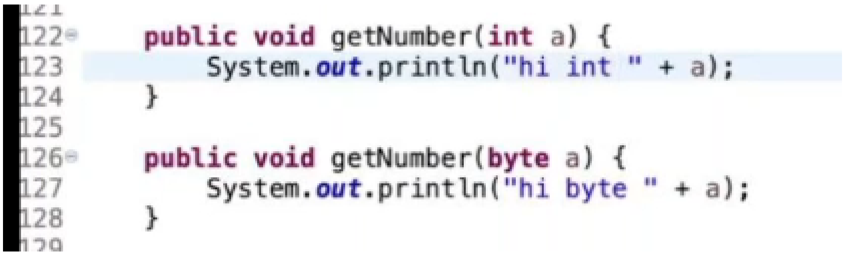
```java

    public static void main(String[] args) {

        Browser obj = new Browser();
        boolean exec = obj.launchBrowser("chrome");
        System.out.println(exec);

        if (exec) {
            System.out.println("enter the URL");
        }

```



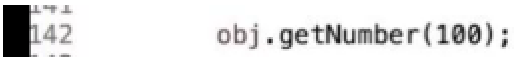
<terminated> Browser (9) [Java Application] /Users

```
browser name : chrome
launch google chrome
true
enter the URL
```

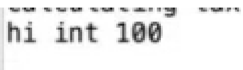
From same families overloading-



```java
    public void getNumber(int a) {
        System.out.println("hi int " + a);
    }

    public void getNumber(byte a) {
        System.out.println("hi byte " + a);
    }
```



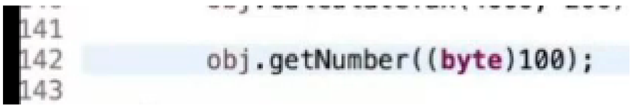
```java
        obj.getNumber(100);
```



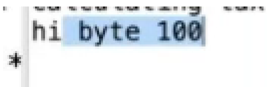
```
hi int 100
```

In java by default all numbers without decimals are int.

To call byte-

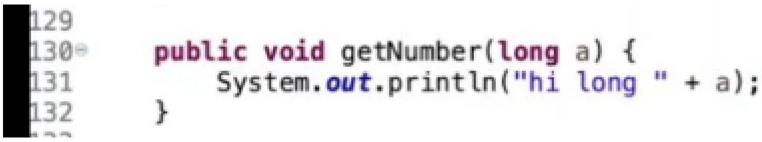


```java
        obj.getNumber((byte)100);
```

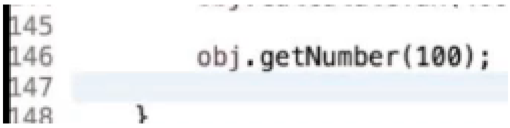


```
hi byte 100
```

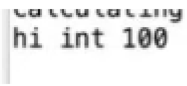
Int versus long-



```java
    public void getNumber(long a) {
        System.out.println("hi long " + a);
    }
```



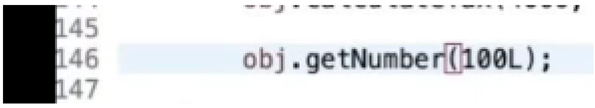
```java
        obj.getNumber(100);

    }
```



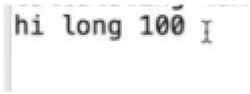
```
hi int 100
```

Call long-

[redacted]



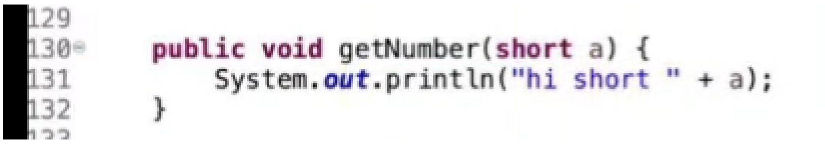
```
        obj.getNumber(100L);
```



```
hi long 100
```

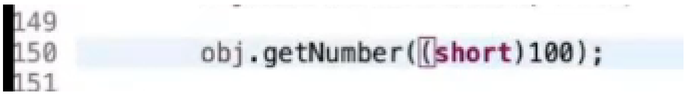
Short-

[redacted]

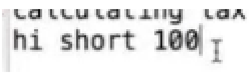


```
    public void getNumber(short a) {
        System.out.println("hi short " + a);
    }
```

[redacted]



```
        obj.getNumber((short)100);
```



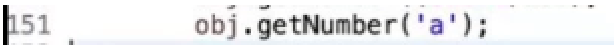
```
calculating tax
hi short 100
```

For decimals double is default.

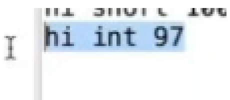
Cast with f and get floating.

Character-

just pass character and we get the ascii value.

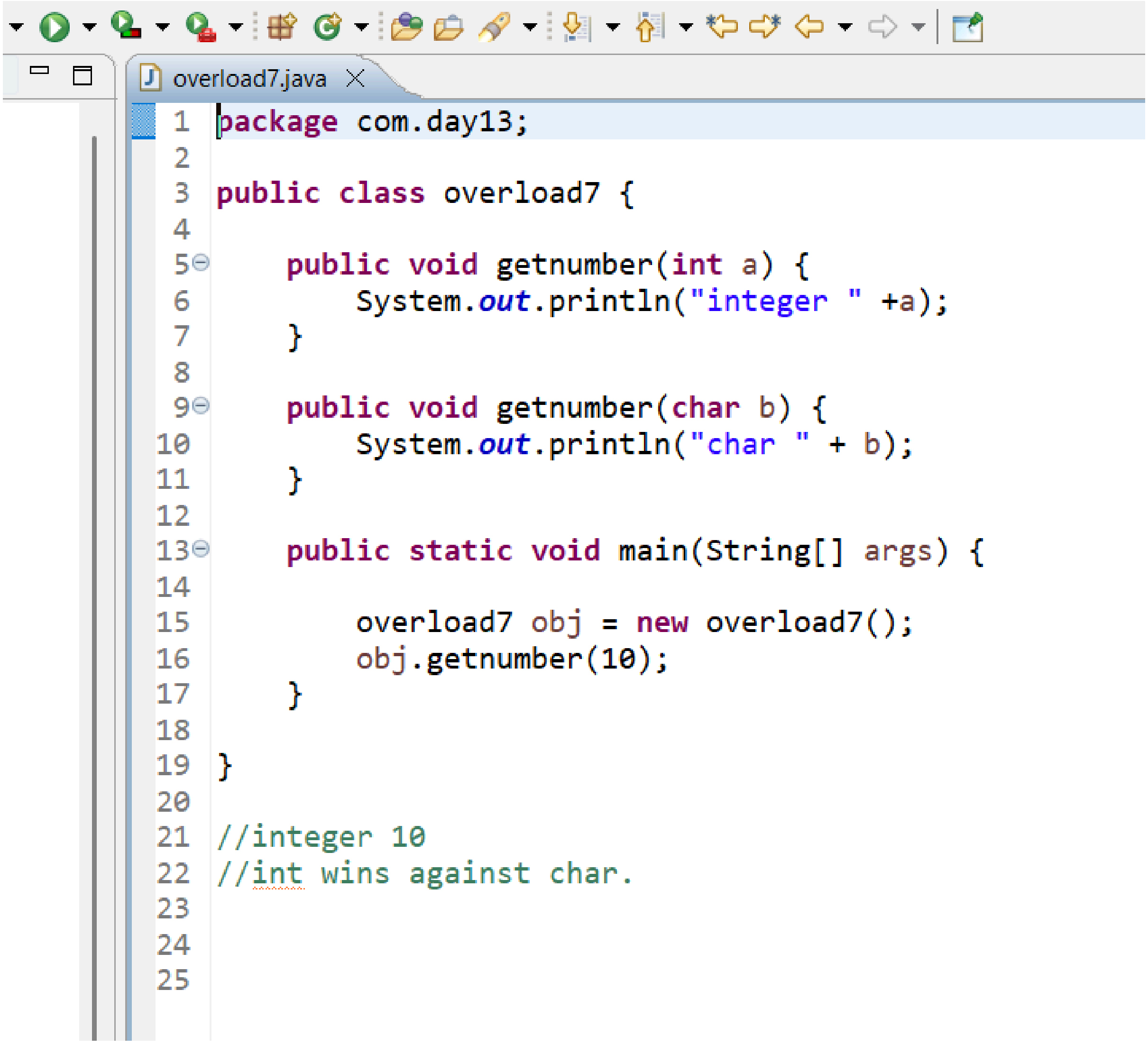


```
        obj.getNumber('a');
```

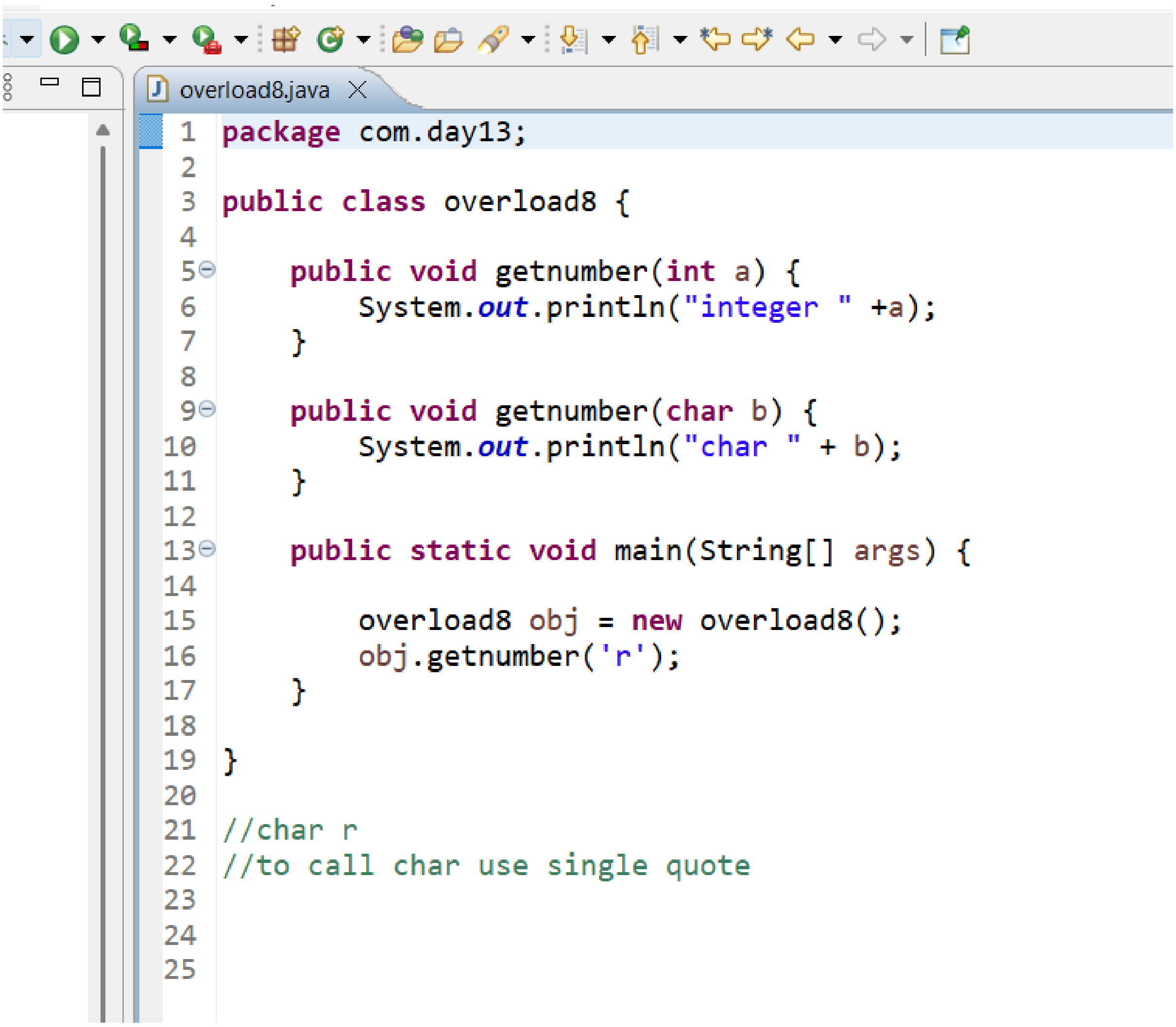


```
hi short 100
hi int 97
```

paste overload7 and overload8-

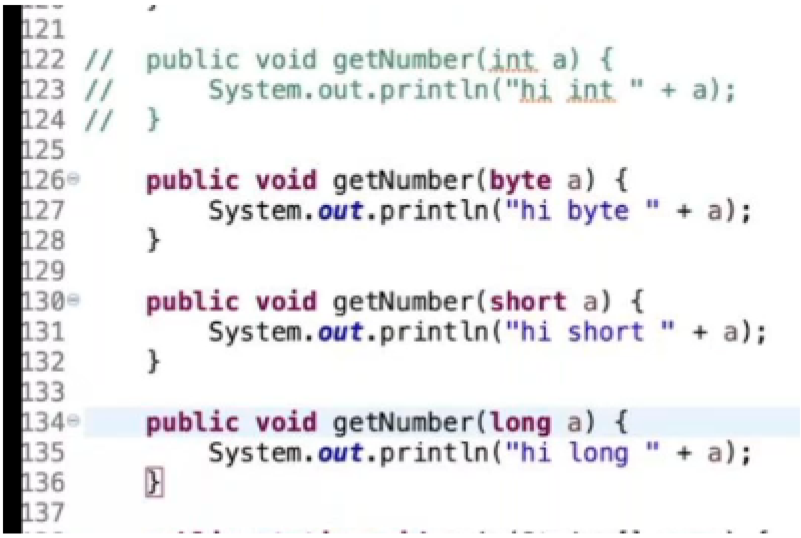


```java
package com.day13;

public class overload7 {

    public void getnumber(int a) {
        System.out.println("integer " +a);
    }

    public void getnumber(char b) {
        System.out.println("char " + b);
    }

    public static void main(String[] args) {

        overload7 obj = new overload7();
        obj.getnumber(10);
    }

}

//integer 10
//int wins against char.
```

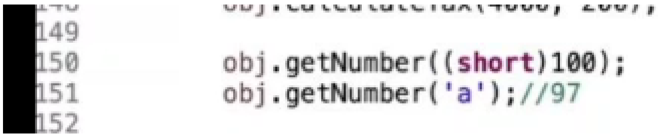


```
package com.day13;

public class overload8 {

    public void getnumber(int a) {
        System.out.println("integer " +a);
    }

    public void getnumber(char b) {
        System.out.println("char " + b);
    }

    public static void main(String[] args) {

        overload8 obj = new overload8();
        obj.getnumber('r');
    }

}

//char r
//to call char use single quote
```

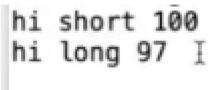
When int method is not there-



```
//  public void getNumber(int a) {
//      System.out.println("hi int " + a);
//  }

    public void getNumber(byte a) {
        System.out.println("hi byte " + a);
    }

    public void getNumber(short a) {
        System.out.println("hi short " + a);
    }

    public void getNumber(long a) {
        System.out.println("hi long " + a);
    }

```

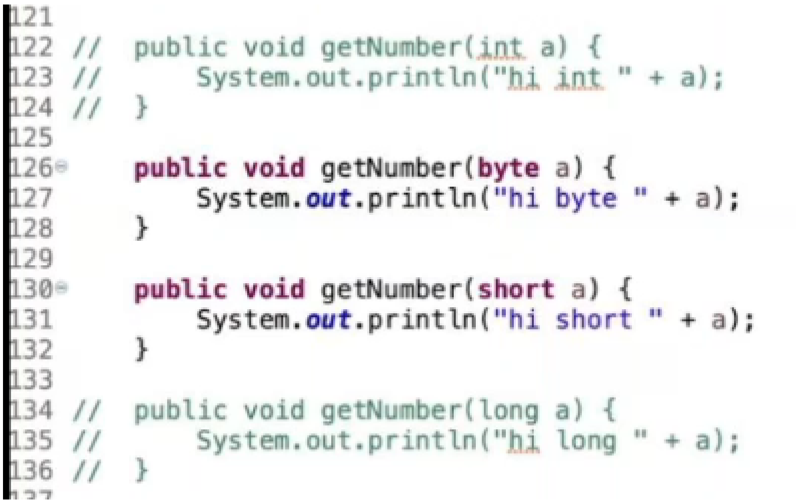


```
        obj.calculateTax(4000, 200);

        obj.getNumber((short)100);
        obj.getNumber('a');//97

```

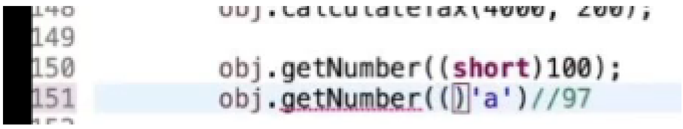


```
hi short 100
hi long 97
```

//second highest precedence is for long after int.

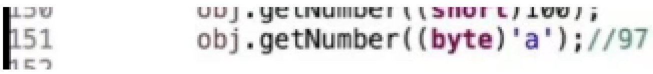


```
//  public void getNumber(int a) {
//      System.out.println("hi int " + a);
//  }

    public void getNumber(byte a) {
        System.out.println("hi byte " + a);
    }

    public void getNumber(short a) {
        System.out.println("hi short " + a);
    }

//  public void getNumber(long a) {
//      System.out.println("hi long " + a);
//  }
```



```
        obj.calculateTax(4000, 200);

        obj.getNumber((short)100);
        obj.getNumber(()'a')//97
```

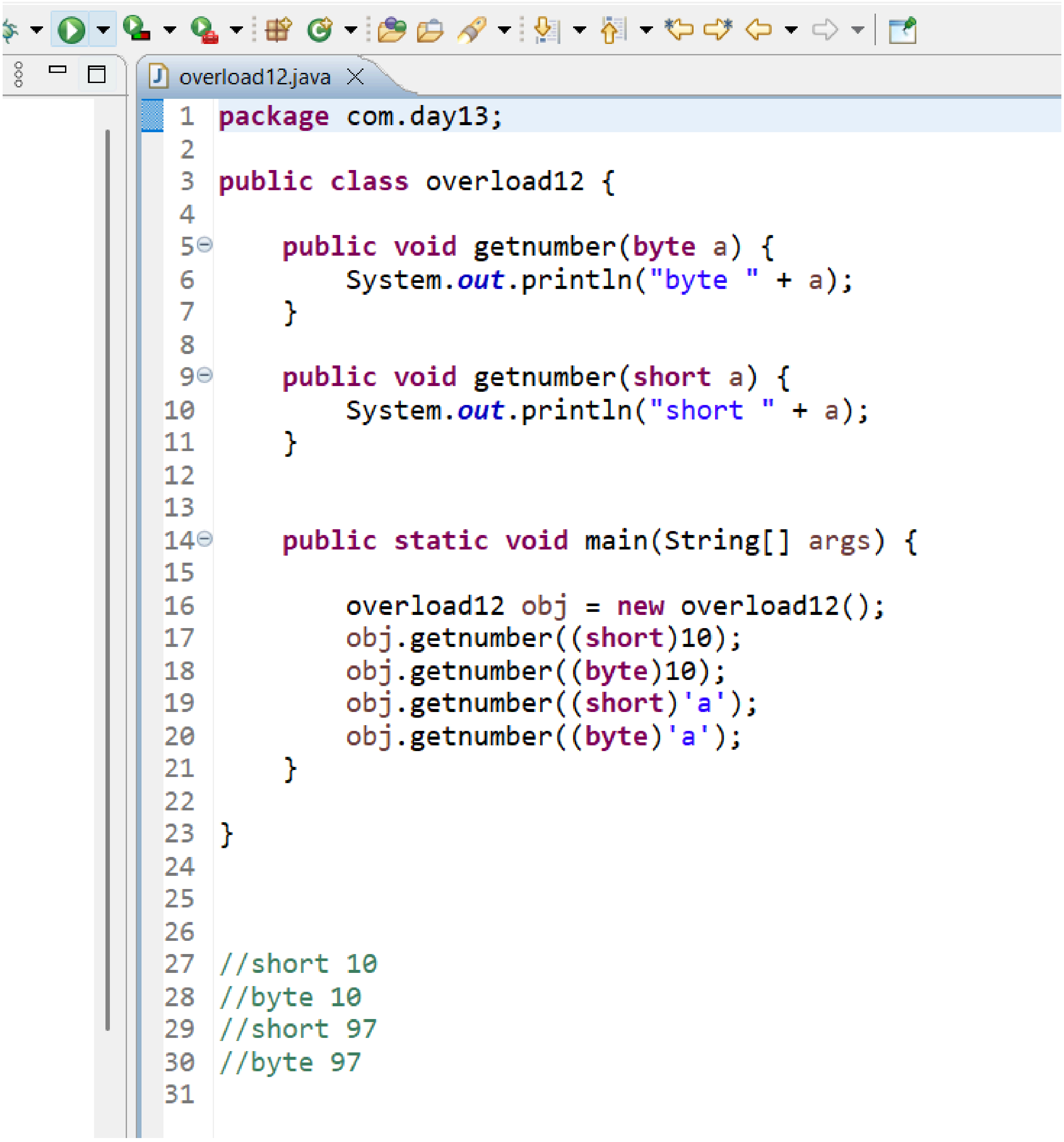
Error for character as no long, no int present.



```
obj.getNumber((short)100);
obj.getNumber((byte)'a');//97
```

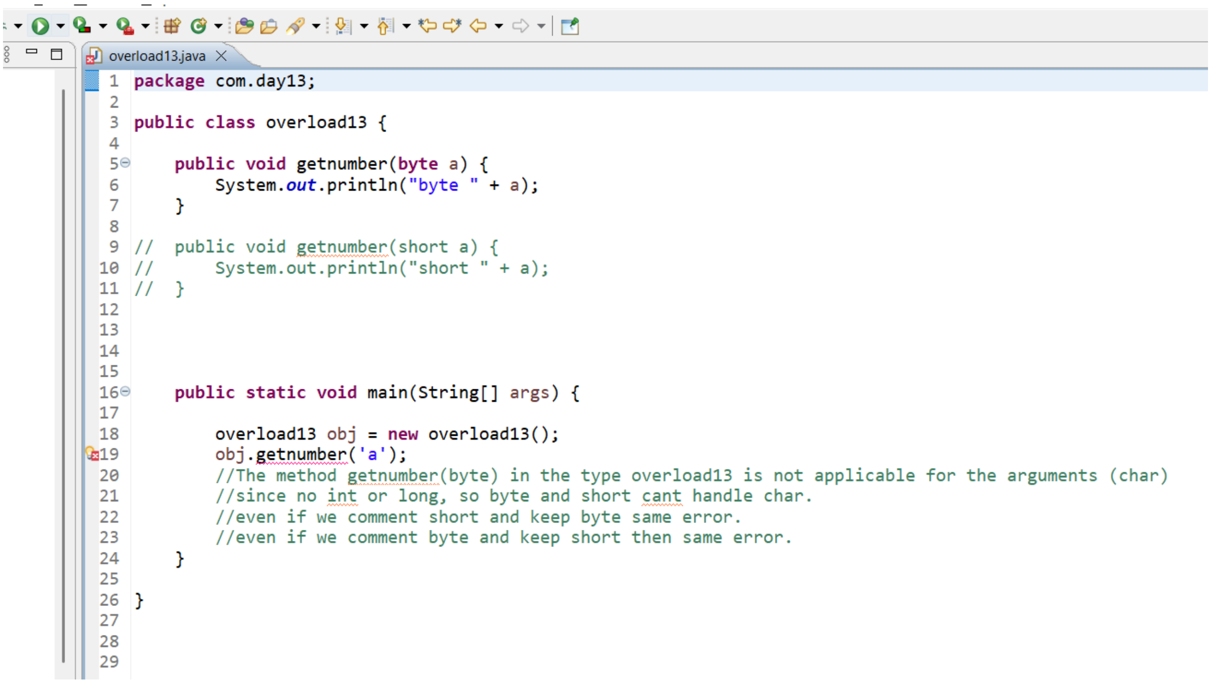
Hi byte 97.

paste overload12



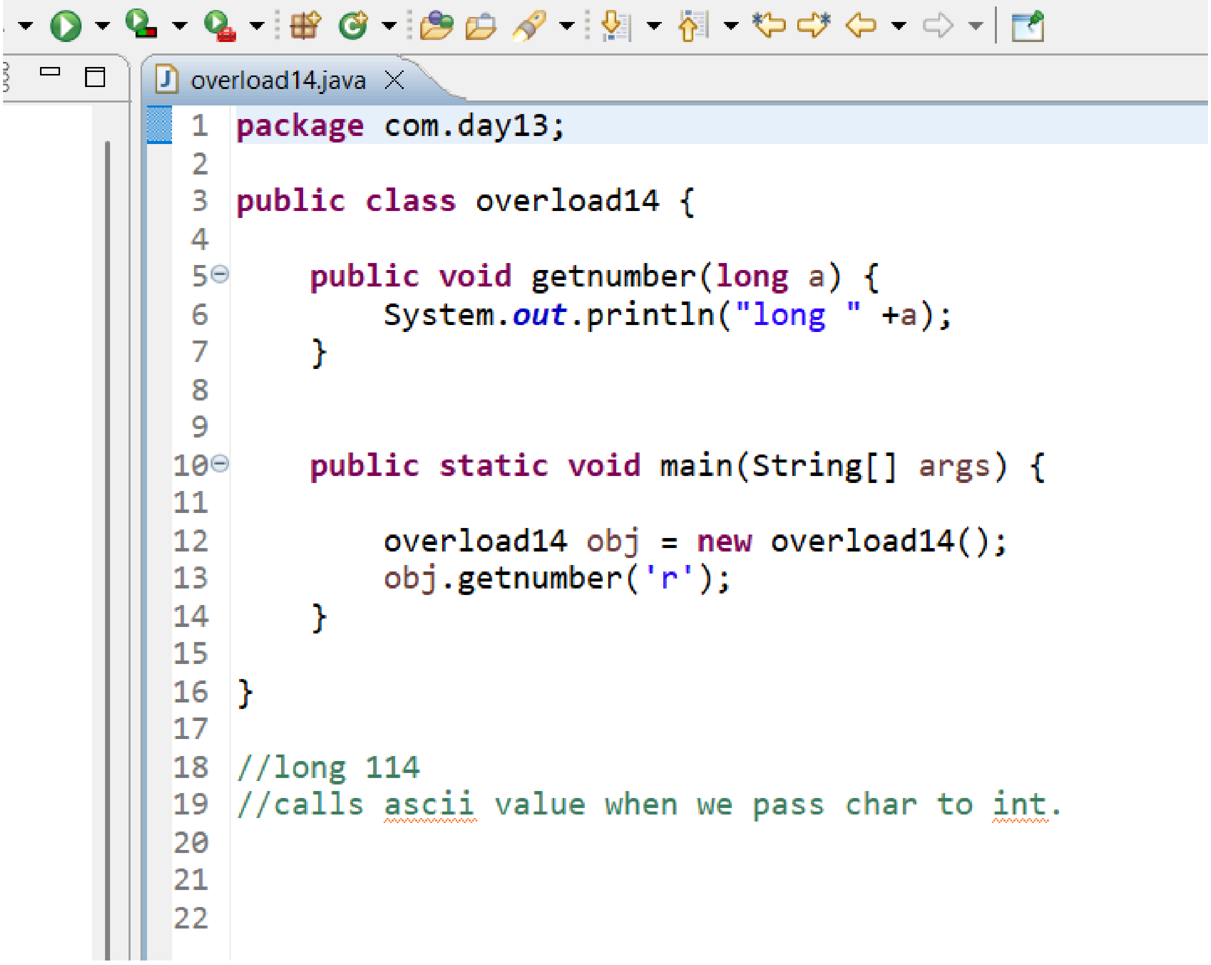
```
package com.day13;

public class overload12 {

    public void getnumber(byte a) {
        System.out.println("byte " + a);
    }

    public void getnumber(short a) {
        System.out.println("short " + a);
    }


    public static void main(String[] args) {

        overload12 obj = new overload12();
        obj.getnumber((short)10);
        obj.getnumber((byte)10);
        obj.getnumber((short)'a');
        obj.getnumber((byte)'a');
    }

}



//short 10
//byte 10
//short 97
//byte 97

```

paste overload 13



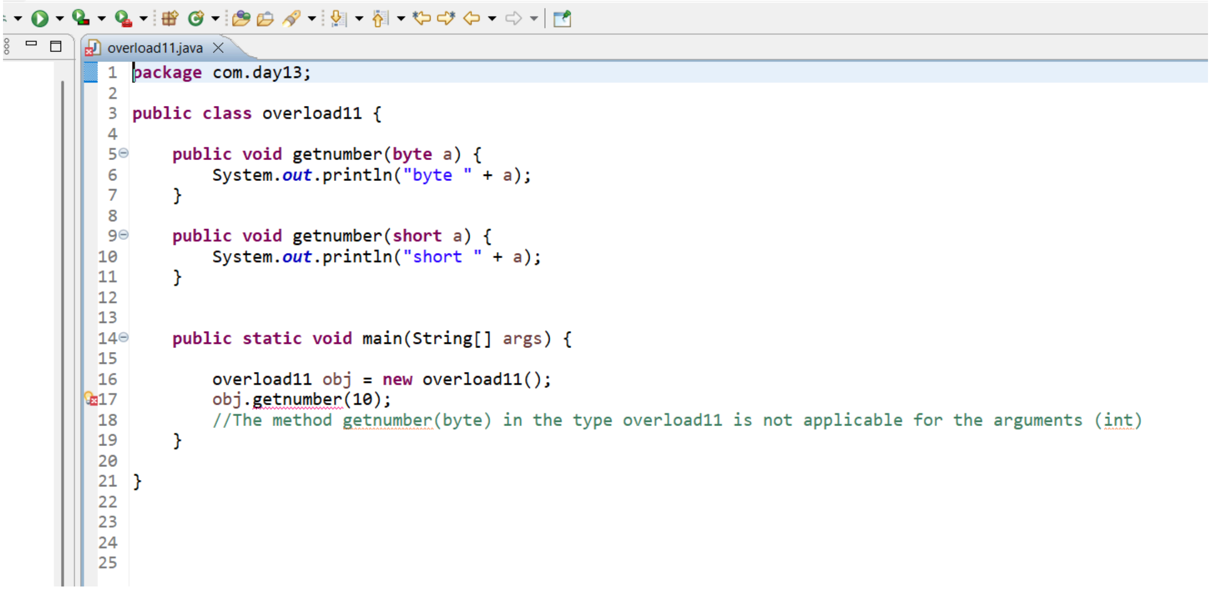
```
package com.day13;

public class overload13 {

    public void getnumber(byte a) {
        System.out.println("byte " + a);
    }

//  public void getnumber(short a) {
//      System.out.println("short " + a);
//  }




    public static void main(String[] args) {

        overload13 obj = new overload13();
        obj.getnumber('a');
        //The method getnumber(byte) in the type overload13 is not applicable for the arguments (char)
        //since no int or long, so byte and short cant handle char.
        //even if we comment short and keep byte same error.
        //even if we comment byte and keep short then same error.
    }

}
```

paste overload 14

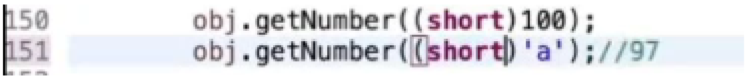


```
package com.day13;

public class overload14 {

    public void getnumber(long a) {
        System.out.println("long " +a);
    }


    public static void main(String[] args) {

        overload14 obj = new overload14();
        obj.getnumber('r');
    }

}

//long 114
//calls ascii value when we pass char to int.
```

paste overload11



```
package com.day13;

public class overload11 {

    public void getnumber(byte a) {
        System.out.println("byte " + a);
    }

    public void getnumber(short a) {
        System.out.println("short " + a);
    }


    public static void main(String[] args) {

        overload11 obj = new overload11();
        obj.getnumber(10);
        //The method getnumber(byte) in the type overload11 is not applicable for the arguments (int)
    }

}
```



```
        obj.getNumber((short)100);
        obj.getNumber((short)'a');//97
```

Hi short 97.