

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

paste the switch1

```
switch1.java
1 package com.day13;
2
3 public class switch1 {
4
5     public int getstudentmarks(String name) {
6         System.out.println("student name is " + name);
7
8         switch (name.toLowerCase().trim()) {
9             case "piyush":
10                 return 80;
11                 break; //Unreachable code
12             case "rose":
13                 return 90;
14             case "tiger":
15                 return 70;
16             default:
17                 System.out.println("pass right student name");
18                 return -1; //best practice to return false or negative boolean when no condition match.
19                 //we can also return zero.
20             }
21     }
22 }
23
24
```

```
7 //0 to 100
8
9
10 public int getStudentMarks(String name) {
11     System.out.println("student name : " + name);
12
13     switch (name.toLowerCase().trim()) {
14         case "priya":
15             return 90;
16             break;
17
18         default:
19             break;
20     }
21 }
22
23
24 public static void main(String[] args) {
```

```

1 package javasessions;
2
3 public class StudentSheet {
4
5     //WAF: getStudentMarks(String name)
6     //return: marks(int)
7     //0 to 100
8
9
10
11     //WAF: getStudentMarks(String name)
12     //return: marks(int)
13     //0 to 100
14     //student is not found : return -1
15
16     public int getStudentMarks(String name) {
17         System.out.println("student name : " + name);
18
19         switch (name.toLowerCase().trim()) {
20             case "priya":
21                 return 90;
22             case "ravi":
23                 return 10;
24             case "shikha":
25                 return 100;
26             default:
27                 System.out.println("plz pass the right student name " + name);
28                 return -1;
29         }
30     }
31 }

```

```

27
28
29     public static void main(String[] args) {
30         StudentSheet sh = new StudentSheet();
31         int m = sh.getStudentMarks("priya");
32         System.out.println(m);
33     }
34 }
35
36 }

```

```

<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

```

duplicate1.java x
1 package com.day13;
2
3 public class duplicate1 {
4
5     //cannot have duplicate method in one class.
6
7     public void test1() {
8
9     }
10
11     public void test1() { //Duplicate method test1() in type duplicate1
12
13     }
14
15     public double test1() { //Duplicate method test1() in type duplicate1
16
17     }
18
19     public void test1(String name) { //not duplicate.
20
21     }
22
23     public void test1(String name) { //Duplicate method test1(String) in type duplicate1
24
25     }
26
27     public double test1(String name) { //Duplicate method test1(String) in type duplicate1
28
29     }
30
31     //so return type doesnt matter when checking for duplicacy, only the parameters matter.
32
33 }
34

```

## Method overloading-

```

1 package javasessions;
2
3 public class EmployeeSheet {
4
5     public void test() { // 0 param
6
7     }
8
9     public void test(int a) { // 1 param
10
11     }
12
13     public void test(String x) { // 1 param
14
15     }
16
17     public void test(String x, String y) { // 2 param
18
19     }
20
21     public void test(String a, int b) { // 2 param
22
23     }
24
25     public void test(int a, String b) { // 2 param
26
27     }
28
29     public static void main(String[] args) {
30
31     }
32
33 }
34

```

Also known as polymorphism.

Poly means many.

Morphism means different forms.

Within same class.

```

Batch 5 //method overloading: Poly(many)+morphism(form)
tAss 6 //within the same class if we have multiple methods:
n 7 //1. with the same method name
vaSe 8 //2. with the different number of parameters
POM 9 //3. with the different sequence of parameters
ileni 10 //4. return type doesn't matter here
rSest 11
n Lib

```

```

14
13= public int test() { // 0 param
14     return 100;
15 }
16
17= public double test(int a) { // 1 param
18     return 12.33;
19 }
20

```

```

5 //method overloading: Poly(many)+morphism(form)
6 //within the same class if we have multiple methods:
7 //1. with the same method name
8 //2. with the different number of parameters
9 //3. with the different sequence of parameters (if number of params are the same)
10 //4. return type doesn't matter here
11

```

```

28
29= public void test(String a, int b) { //2 param
30 }
31
32
33= public int test(int a, String b) { //2 param
34     return 100;
35 }
36

```

```

38
39= public static void main(String[] args) {
40
41     EmployeeSheet obj = new EmployeeSheet();
42     obj.test(90);
43 }
44

```

Test(int a) method called.

Compiler knows which method to call.

```

5 //method overloading: Poly(many)+morphism(form): compileTime(static)
6 //within the same class if we have multiple methods:
7 //1. with the same method name
8 //2. with the different number of parameters
9 //3. with the different sequence of parameters (if number of params are the same)
10 //4. return type doesn't matter here
11

```



```

54
55 // search:
56 public void search() {
57
58 }
59
60 public void search(String name) {
61
62 }
63
64 public void search(String name, String color) {
65
66 }
67
68 public void search(String name, String color, int price) {
69     System.out.println("search with: " + name + " " + color + " " + price);
70 }
71

```

```

72
73 public static void main(String[] args) {
74
75     EmployeeSheet obj = new EmployeeSheet();
76     obj.test("naveen", 70);
77
78     obj.search("Tshirt", "Black", 1000);
79

```

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

```

71
72 // payment:
73 public void doPayment(String upi) {
74
75 }
76
77 public void doPayment(String cc, int cvv) {
78
79 }
80
81 public void doPayment(String cc, int cvv, int otp) {
82
83 }
84

```



```

85 // uber:
86 // carBooking:
87 public void carBooking(String stPoint, String endPoint) {
88
89 }
90
91 public void carBooking(String carType, String stPoint, String endPoint) {
92
93 }
94
95 public void carBooking(String carType, String stPoint, String endPoint, int passengers) {
96
97 }
98

```

```

103 //
104 public double calculateTax(int totalIncome, int bonus, int stocksProfit) {
105     System.out.println("calculating tax");
106     double totalTax = (totalIncome/30)*100 + (bonus/5)*100;
107     return totalTax;
108 }
109

```

```

119
120     double tax = obj.calculateTax(1000, 500, 300);
121     System.out.println(tax);
122 }

```

13300.00.

```

102
103 //
104 public double calculateTax(int totalIncome, int bonus, int stocksProfit) {
105     System.out.println("calculating tax");
106     double totalTax = (totalIncome*30)/100 + (bonus*5)/100 + (stocksProfit*2)/100;
107     return totalTax;
108 }
109
110 public double calculateTax(int totalIncome, int bonus) {
111     System.out.println("calculating tax");
112     double totalTax = (totalIncome*30)/100 + (bonus*5)/100;
113     return totalTax;
114 }
115
131
132     double tax = obj.calculateTax(1000, 500, 300);
133     System.out.println(tax);
134
135
136     obj.calculateTax(4000, 200);
137

```

Note-

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



## Void return-

paste void1

```

void1.java X
1 package com.day13;
2
3 public class void1 {
4
5     //when void is return then blank return also works.
6
7     public void click(String element) {
8         System.out.println("click on the element " + element);
9         return;
10    }
11
12    public static void main(String[] args) {
13
14        void1 v1=new void1();
15        v1.click("login");
16    }
17
18 }
19
20 //click on the element login
21
22

```

```

FunctionsInJava.java StudentSheet.java Customer.java EmployeeSheet.java Browser.java X
1 package javasessions;
2
3 public class Browser {
4
5
6     public void click(String element) {
7         System.out.println("click on element " + element);
8     }
9
10    public void sendKeys(String element, String value) {
11        System.out.println("entering value : " + value + " into the element : " + element);
12    }
13
14    public String getTitle() {
15        return "google";
16    }
17
18

```

```

18
19 //WAF: launchBrowser(String browserName)
20 //browserName: chrome, firefox, safari, edge
21 //return: print: success messg and return boolean(true/false)
22

```

Boolean not allowed in switch() –

paste browser2-

```

1 package com.day13;
2
3 public class browser2 {
4
5     public boolean launchBrowser(String browserName) {
6
7         System.out.println("browser name : " + browserName);
8         boolean flag = true;
9
10        //switch cannot have boolean as its parameter.
11        switch (flag) { //Cannot switch on a value of type boolean. Only convertible int
12        // values, strings or enum variables are permitted
13            case "chrome":
14                System.out.println("launch google chrome");
15                break;
16
17            case "firefox":
18                System.out.println("launch firefox");
19                break;
20
21            case "edge":
22                System.out.println("launch edge");
23                break;
24
25            case "safari":
26                System.out.println("launch safari");
27                break;
28
29            default:
30                System.out.println("plz pass the right browser name : " + browserName);
31                break;
32        }
33
34        return flag;
35    }
36
37 }

```

```

26
27 switch (browserName.toLowerCase().trim()) {

```

```

22
23 public boolean launchBrowser(String browserName) {
24
25     System.out.println("browser name : " + browserName);
26     boolean flag = true;
27
28     switch (browserName.toLowerCase().trim()) {
29         case "chrome":
30             System.out.println("launch google chrome");
31             break;
32         case "firefox":
33             System.out.println("launch firefox");
34             break;
35         case "edge":
36             System.out.println("launch edge");
37             break;
38         case "safari":
39             System.out.println("launch safari");
40             break;
41
42         default:
43             System.out.println("plz pass the right browser name : " + browserName);
44             break;
45     }
46
47     return flag;
48 }
49
50
51
52
53 public static void main(String[] args) {
54
55
56     Browser obj = new Browser();
57     boolean exec = obj.launchBrowser("chrome");
58     System.out.println(exec);
59
60
61 }
62
63 }
64

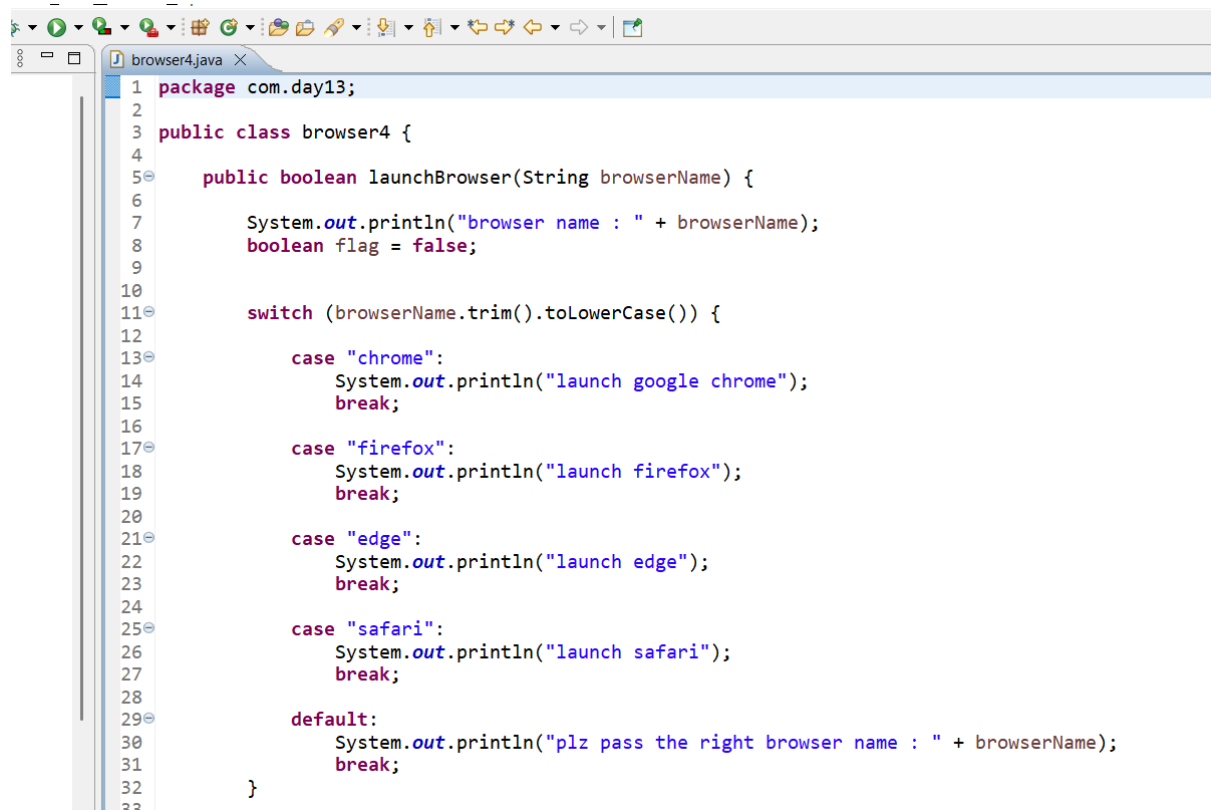
```

```

<terminated> Browser (9) [Java Application] /User
browser name : chrome
launch google chrome
true

```

paste browser4



```
1 package com.day13;
2
3 public class browser4 {
4
5     public boolean launchBrowser(String browserName) {
6
7         System.out.println("browser name : " + browserName);
8         boolean flag = false;
9
10
11         switch (browserName.trim().toLowerCase()) {
12
13             case "chrome":
14                 System.out.println("launch google chrome");
15                 break;
16
17             case "firefox":
18                 System.out.println("launch firefox");
19                 break;
20
21             case "edge":
22                 System.out.println("launch edge");
23                 break;
24
25             case "safari":
26                 System.out.println("launch safari");
27                 break;
28
29             default:
30                 System.out.println("plz pass the right browser name : " + browserName);
31                 break;
32         }
33     }
```

```

30         System.out.println("Please pass the flag");
31         break;
32     }
33
34     return flag;
35 }
36
37 public static void main(String[] args) {
38
39     browser4 b3=new browser4();
40     boolean exec=b3.launchBrowser("chrome");
41     System.out.println(exec);
42 }
43
44 }
45
46 //browser name : chrome
47 //launch google chrome
48 //false
49
50
51

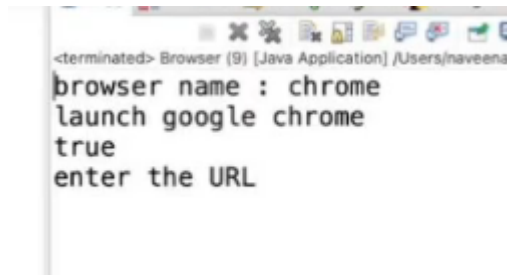
```

able Smart Insert 1:1:0

```

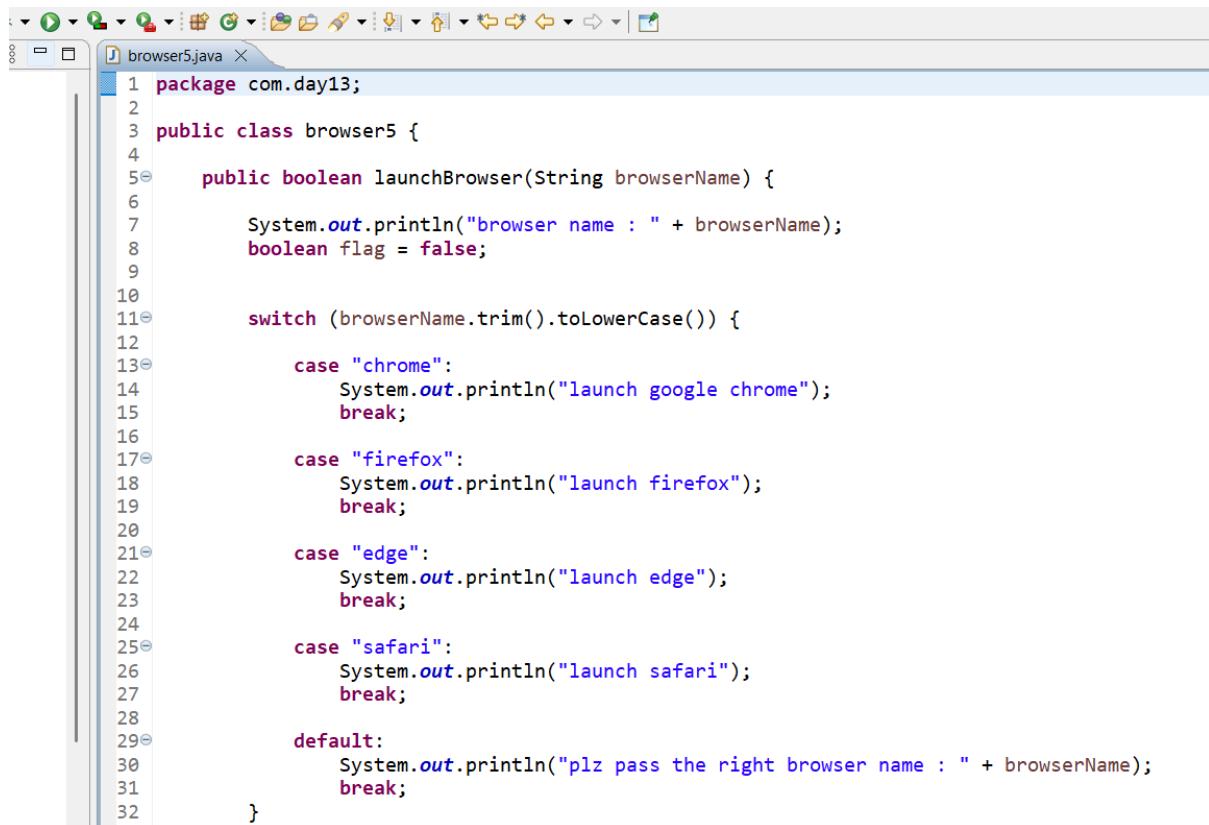
46
47 }
48
49 public static void main(String[] args) {
50
51     Browser obj = new Browser();
52     boolean exec = obj.launchBrowser("safari");
53     System.out.println(exec);
54
55     if (exec) {
56         System.out.println("enter the URL");
57     }
58
59 }

```



```
<terminated> Browser (9) [Java Application] /Users/haveena
browser name : chrome
launch google chrome
true
enter the URL
```

paste browser5



```
1 package com.day13;
2
3 public class browser5 {
4
5     public boolean launchBrowser(String browserName) {
6
7         System.out.println("browser name : " + browserName);
8         boolean flag = false;
9
10
11         switch (browserName.trim().toLowerCase()) {
12
13             case "chrome":
14                 System.out.println("launch google chrome");
15                 break;
16
17             case "firefox":
18                 System.out.println("launch firefox");
19                 break;
20
21             case "edge":
22                 System.out.println("launch edge");
23                 break;
24
25             case "safari":
26                 System.out.println("launch safari");
27                 break;
28
29             default:
30                 System.out.println("plz pass the right browser name : " + browserName);
31                 break;
32         }
33     }
34 }
```

```

30         System.out.println( p12 pass the right browser name
31         break;
32     }
33
34     return flag;
35 }
36
37 public static void main(String[] args) {
38
39     browser5 b3=new browser5();
40     boolean exec=b3.launchBrowser("chrome");
41     System.out.println(exec);
42
43     if(exec) {
44         System.out.println("run the url");
45     }
46 }
47
48 }
49
50 //browser name : chrome
51 //launch google chrome
52 //false
53
54
55

```

Wrong handling of flag-

Even with invalid browser, it gives flag true.

```

47     }
48
49     public static void main(String[] args) {
50
51         Browser obj = new Browser();
52         boolean exec = obj.launchBrowser("ie");
53         System.out.println(exec);
54
55         if (exec) {
56             System.out.println("enter the URL");
57         }
58
59     }
60

```



```

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

```

## Proper way to use flags for invalid case-

```

39
40     default:
41         System.out.println("plz pass the right browser name : " + browserName);
42         flag = false;
43         break;
44     }
45
46     return flag;

```

```

47
48     }
49
50     public static void main(String[] args) {
51
52         Browser obj = new Browser();
53         boolean exec = obj.launchBrowser("ie");
54         System.out.println(exec);
55
56         if (exec) {
57             System.out.println("enter the URL");
58         }
59
60     }

```

```

<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.

```

21- public boolean launchBrowser(String browserName) {
22-
23-     System.out.println("browser name : " + browserName);
24-     boolean flag = false;
25-
26-     switch (browserName.toLowerCase().trim()) {
27-     case "chrome":
28-         System.out.println("launch google chrome");
29-         flag = true;
30-         break;
31-     case "firefox":
32-         System.out.println("launch firefox");
33-         flag = true;
34-         break;
35-     case "edge":
36-         System.out.println("launch edge");
37-         flag = true;
38-         break;
39-     case "safari":
40-         System.out.println("launch safari");
41-         flag = true;
42-         break;
43-
44-     default:
45-         System.out.println("plz pass the right browser name : " + browserName);
46-         //flag = false;
47-         break;
48-     }
49-
50-     return flag;
51- }
52-
53-

```

```

53-
54- public static void main(String[] args) {
55-
56-     Browser obj = new Browser();
57-     boolean exec = obj.launchBrowser("chrome");
58-     System.out.println(exec);
59-
60-     if (exec) {
61-         System.out.println("enter the URL");
62-     }
63-

```

```

<terminated> Browser (9) [Java Application] /Users
browser name : chrome
launch google chrome
true
enter the URL

```

From same families overloading-

```

122 public void getNumber(int a) {
123     System.out.println("hi int " + a);
124 }
125
126 public void getNumber(byte a) {
127     System.out.println("hi byte " + a);
128 }
129
141
142     obj.getNumber(100);

```

hi int 100

In java by default all numbers without decimals are int.

To call byte-

```

141
142     obj.getNumber((byte)100);
143

```

hi byte 100

Int versus long-

```

129
130 public void getNumber(long a) {
131     System.out.println("hi long " + a);
132 }
133
145
146     obj.getNumber(100);
147 }
148

```

hi int 100

Call long-

```

145
146 obj.getNumber(100L);
147
hi long 100

```

## Short-

```

129
130 public void getNumber(short a) {
131     System.out.println("hi short " + a);
132 }
133
149
150 obj.getNumber((short)100);
151
calculating tax
hi short 100

```

For decimals double is default.

Cast with f and get floating.

## Character-

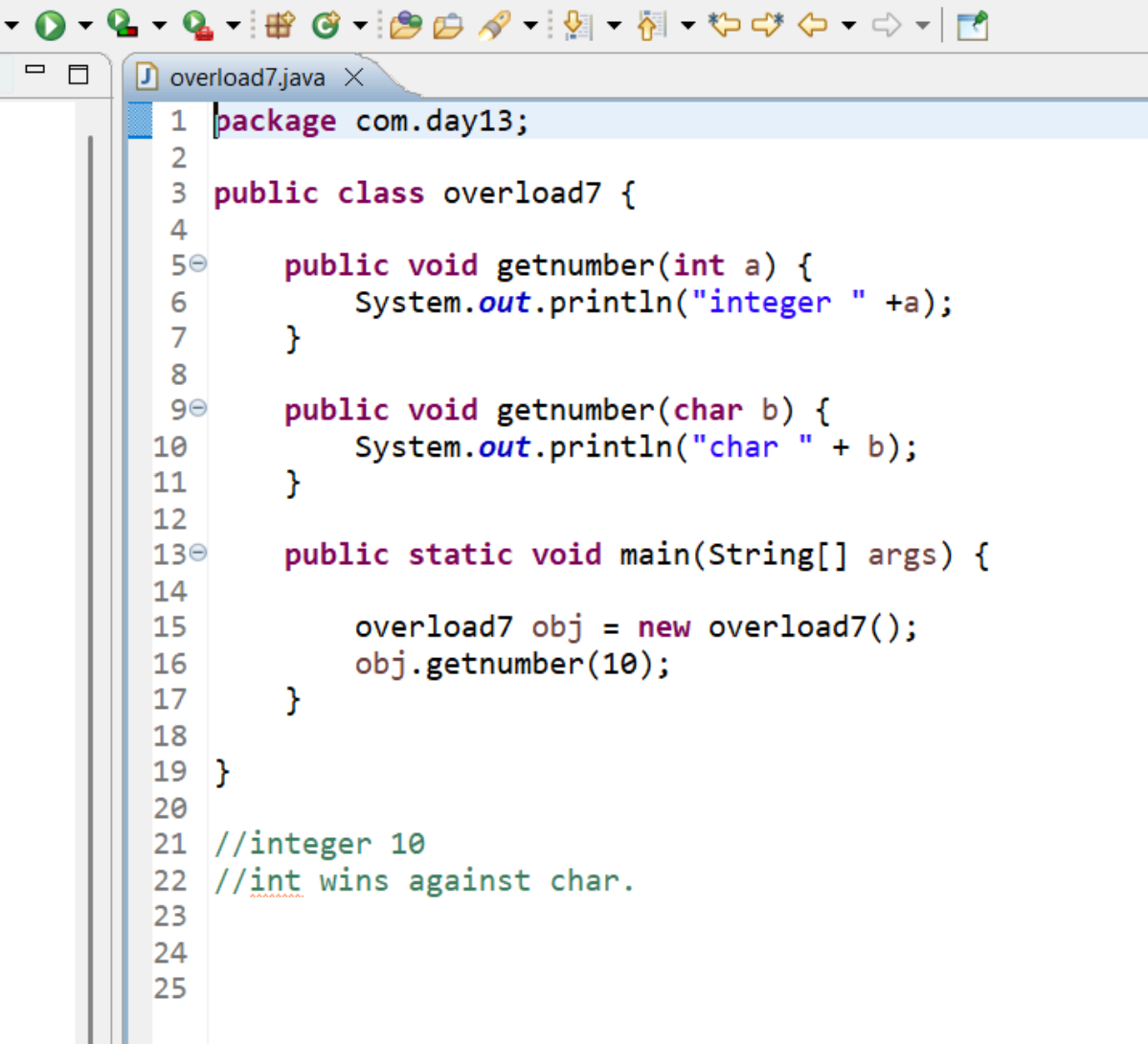
just pass character and we get the ascii value.

```

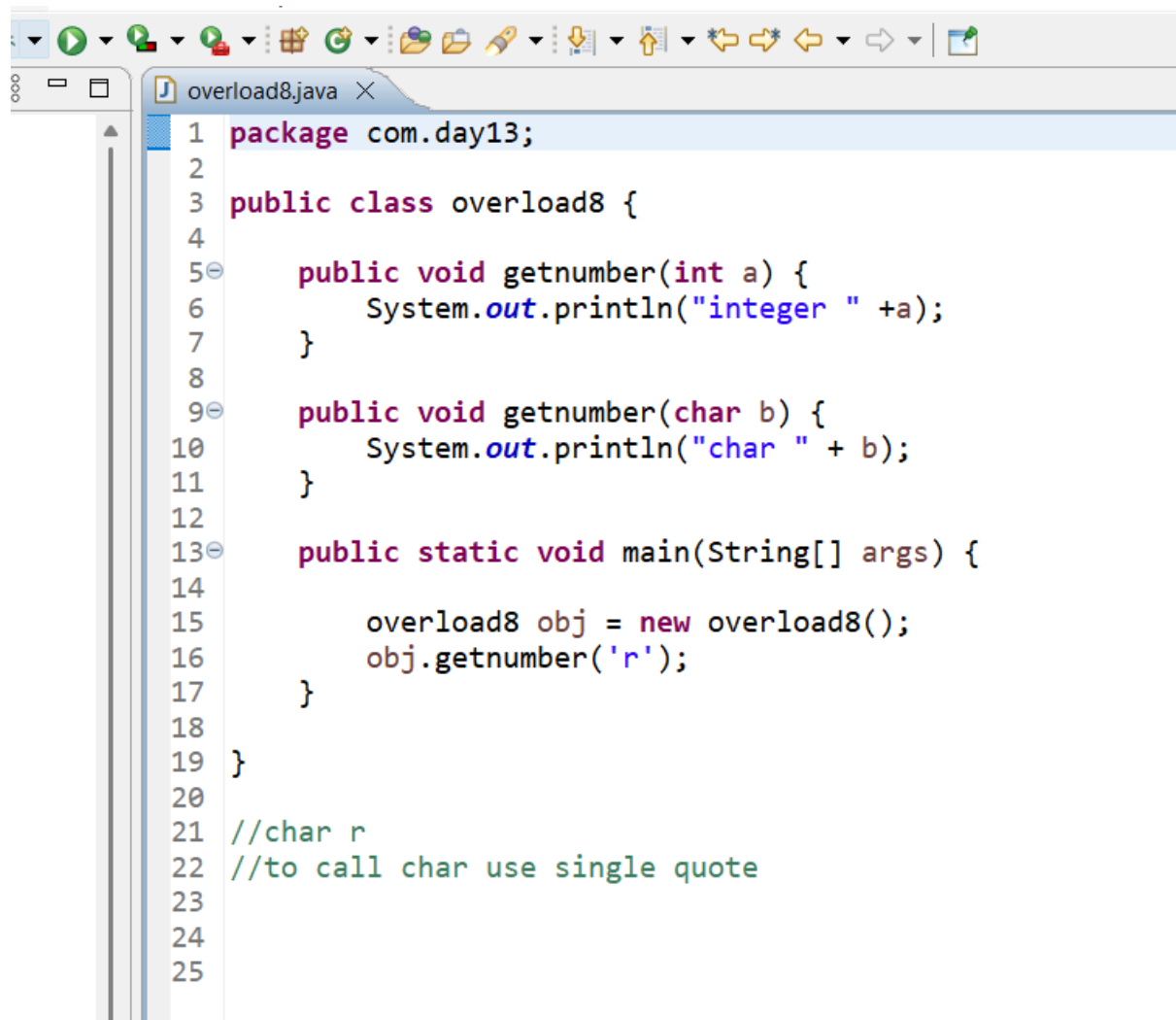
151 obj.getNumber('a');
hi short 100
hi int 97

```

paste overload7 and overload8-



```
1 package com.day13;
2
3 public class overload7 {
4
5     public void getnumber(int a) {
6         System.out.println("integer " + a);
7     }
8
9     public void getnumber(char b) {
10        System.out.println("char " + b);
11    }
12
13    public static void main(String[] args) {
14
15        overload7 obj = new overload7();
16        obj.getnumber(10);
17    }
18 }
19
20
21 //integer 10
22 //int wins against char.
23
24
25
```



```
1 package com.day13;
2
3 public class overload8 {
4
5     public void getnumber(int a) {
6         System.out.println("integer " + a);
7     }
8
9     public void getnumber(char b) {
10        System.out.println("char " + b);
11    }
12
13    public static void main(String[] args) {
14
15        overload8 obj = new overload8();
16        obj.getnumber('r');
17    }
18
19 }
20
21 //char r
22 //to call char use single quote
23
24
25
```

When int method is not there-

```

121
122 // public void getNumber(int a) {
123 //     System.out.println("hi int " + a);
124 // }
125
126 public void getNumber(byte a) {
127     System.out.println("hi byte " + a);
128 }
129
130 public void getNumber(short a) {
131     System.out.println("hi short " + a);
132 }
133
134 public void getNumber(long a) {
135     System.out.println("hi long " + a);
136 }
137

```

```

148 obj.calculateTax(4000, 200);
149
150 obj.getNumber((short)100);
151 obj.getNumber('a');//97
152

```

```

hi short 100
hi long 97

```

//second highest precedence is for long after int.

```

121
122 // public void getNumber(int a) {
123 //     System.out.println("hi int " + a);
124 // }
125
126 public void getNumber(byte a) {
127     System.out.println("hi byte " + a);
128 }
129
130 public void getNumber(short a) {
131     System.out.println("hi short " + a);
132 }
133
134 // public void getNumber(long a) {
135 //     System.out.println("hi long " + a);
136 // }
137

```

```

148 obj.calculateTax(4000, 200);
149
150 obj.getNumber((short)100);
151 obj.getNumber('a');//97
152

```

Error for character as no long, no int present.



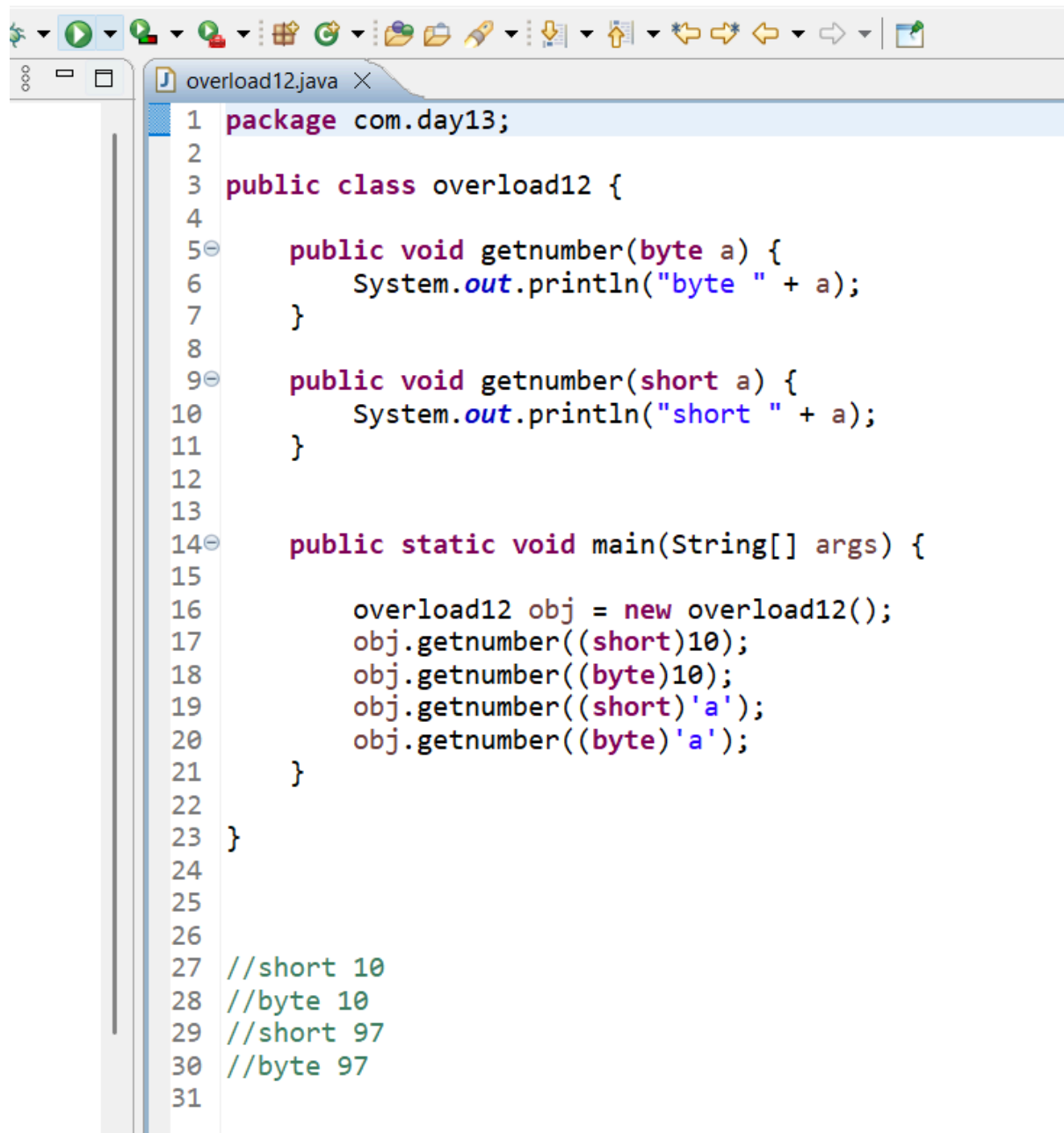
```

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

```

Hi byte 97.

paste overload12



```

1 package com.day13;
2
3 public class overload12 {
4
5     public void getnumber(byte a) {
6         System.out.println("byte " + a);
7     }
8
9     public void getnumber(short a) {
10        System.out.println("short " + a);
11    }
12
13
14    public static void main(String[] args) {
15
16        overload12 obj = new overload12();
17        obj.getNumber((short)10);
18        obj.getNumber((byte)10);
19        obj.getNumber((short)'a');
20        obj.getNumber((byte)'a');
21    }
22
23 }
24
25
26
27 //short 10
28 //byte 10
29 //short 97
30 //byte 97
31

```

paste overload 13

```

1 package com.day13;
2
3 public class overload13 {
4
5     public void getnumber(byte a) {
6         System.out.println("byte " + a);
7     }
8
9     // public void getnumber(short a) {
10    //     System.out.println("short " + a);
11    // }
12
13
14
15
16    public static void main(String[] args) {
17
18        overload13 obj = new overload13();
19        obj.getnumber('a');
20        //The method getnumber(byte) in the type overload13 is not applicable for the arguments (char)
21        //since no int or long, so byte and short cant handle char.
22        //even if we comment short and keep byte same error.
23        //even if we comment byte and keep short then same error.
24    }
25
26 }
27
28
29

```

## paste overload 14

```

1 package com.day13;
2
3 public class overload14 {
4
5     public void getnumber(long a) {
6         System.out.println("long " + a);
7     }
8
9
10    public static void main(String[] args) {
11
12        overload14 obj = new overload14();
13        obj.getnumber('r');
14    }
15
16 }
17
18 //long 114
19 //calls ascii value when we pass char to int.
20
21
22

```

## paste overload11

```

1 package com.day13;
2
3 public class overload11 {
4
5     public void getnumber(byte a) {
6         System.out.println("byte " + a);
7     }
8
9     public void getnumber(short a) {
10        System.out.println("short " + a);
11    }
12
13
14    public static void main(String[] args) {
15
16        overload11 obj = new overload11();
17        obj.getnumber(10);
18        //The method getnumber(byte) in the type overload11 is not applicable for the arguments (int)
19    }
20
21 }
22
23
24
25

```

```

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

```

Hi short 97.