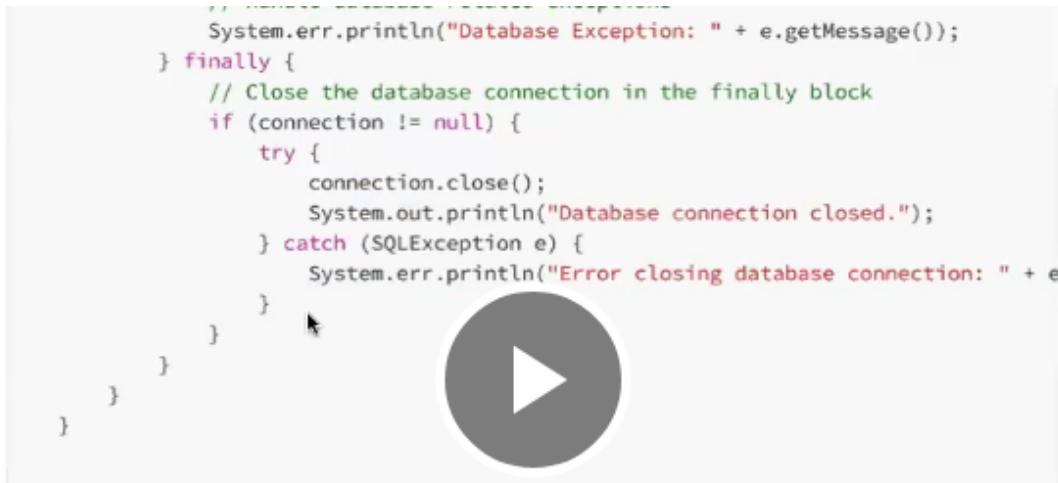


## Anything can come in try catch finally block-



Finally block always runs irrespective of exception.

```
~~
26     //use case:
27
28     //1. make the connection DB: un, pwd, ip:port: try-catch
29     //2. hit the SQL query: try-catch
30     //3. result from DB: try-catch
31     //4. display the data on UI: try-catch
32     //5. logout
33
34     //finally{
35         //6. close connection with DB
36         //}
37
38
39     //reading from excel/file
40
41
```



```

1 package ExceptionHandling;
2
3 public class TestingFinally {
4
5     public static int getMarks(String studentName) {
6
7         System.out.println("getting marks for : " + studentName);
8
9         if (studentName.equals("vibha")) {
10             return 100;
11         }
12
13         else if (studentName.equals("monika")) {
14             return 90;
15         }
16
17         else if (studentName.equals("veena")) {
18             return 80;
19         }
20
21         else {
22             System.out.println("no student found...");
23             return -1;
24         }
25     }
26
27     public static void main(String[] args) {
28
29         int m1 = TestingFinally.getMarks("vibha");
30         System.out.println(m1);
31
32     }
33
34 }
35
36

```

Sessions/src/ExceptionHandling/TestingFinally.java - Eclipse IDE

Console Problems Javadoc Declaration Results of run

<terminated> TestingFinally [Java Application] /Users/naveenautomationlabs/p2

getting marks for : vibha  
100

Add exception-

```

6
7     System.out.println("getting marks for : " + studentName);
8
9     if (studentName.equals("vibha")) {
10        try {
11            int i = 9 / 0;
12        } catch (ArithmaticException e) {
13            System.out.println("AE is coming..");
14        }
15        return 100;
16    }

```

JavaSessions/src/E�ptionHandling/TestingFinally.java - Eclipse IDE

Console Problems Javadoc Declaration

<terminated> TestingFinally [Java Application] /Users/naveenautomat...

getting marks for : vibha  
AE is coming..  
100

paste testfinally1-

```

1 package com.day24;
2
3 public class testfinally1 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10        System.out.println("getting marks for "+studentname);
11        if(studentname.equals("johny")) {
12            //add exception with return.
13            try {
14                int i=9/0;
15            } catch (ArithmaticException e) {
16                System.out.println("arithmetic exception .....");
17            }
18            return 100;
19        }else if(studentname.equals("rose")) {
20            return 90;
21        }else if(studentname.equals("tiger")) {
22            return 80;
23        }else {
24            System.out.println("no student found .....");
25            return -1;
26        }
27
28    //    System.out.println("end of the method");//Unreachable code
29    }

```

```
27  
28 //      System.out.println("end of the method");//Unreachable code  
29 }  
30  
31@ public static void main(String[] args) {  
32  
33     int marks=getmarks("johny");  
34     System.out.println(marks);  
35  
36     int marks1=testfinally1.getmarks("rose");  
37     System.out.println(marks1);  
38  
39     System.out.println("end of the program");  
40 }  
41  
42 }  
43 }  
44  
45 //getting marks for johny  
46 //arithmetic exception .....  
47 //100  
48 //getting marks for rose  
49 //90  
50 //end of the program  
51  
52  
53
```

paste testfinally2-

```

1 package com.day24;
2
3 public class testfinally2 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         System.out.println("getting marks for "+studentname);
11         if(studentname.equals("johny")) {
12             //return inside try lets see.
13             try {
14                 int i=9/0;
15                 return 234234;
16             } catch (ArithmaticException e) {
17                 System.out.println("arithmetic exception .....");
18             }
19             return 100;
20         }else if(studentname.equals("rose")) {
21             return 90;
22         }else if(studentname.equals("tiger")) {
23             return 80;
24         }else {
25             System.out.println("no student found .....");
26             return -1;
27         }
28
29 //        System.out.println("end of the method");//Unreachable code
30
31
32     public static void main(String[] args) {
33
34         int marks=getmarks("johny");
35         System.out.println(marks);
36
37         int marks1=testfinally2.getmarks("rose");
38         System.out.println(marks1);
39
40         System.out.println("end of the program");
41
42     }
43
44 }
45
46 //getting marks for johny
47 //arithmetic exception .....
48 //100
49 //getting marks for rose
50 //90
51 //end of the program
52

```

paste testfinally3-

```

1 package com.day24;
2
3 public class testfinally3 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         System.out.println("getting marks for "+studentname);
11         if(studentname.equals("johny")) {
12             //return inside try lets see.
13             //no error.
14             try {
15                 int i=9/3;
16                 return 234234;
17             } catch (ArithmaticException e) {
18                 System.out.println("arithmetic exception ..... ");
19             }
20             return 100;
21         }else if(studentname.equals("rose")) {
22             return 90;
23         }else if(studentname.equals("tiger")) {
24             return 80;
25         }else {
26             System.out.println("no student found ..... ");
27             return -1;
28         }
29
30         //      System.out.println("end of the method");//Unreachable code
31     }
32
33     public static void main(String[] args) {
34
35         int marks=getmarks("johny");
36         System.out.println(marks);
37
38         int marks1=testfinally3.getmarks("rose");
39         System.out.println(marks1);
40
41         System.out.println("end of the program");
42
43     }
44
45 }
46
47 //getting marks for johny
48 //234234
49 //getting marks for rose
50 //90
51 //end of the program
52

```

paste testfinally4-

```

1 package com.day24;
2
3 public class testfinally4 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         System.out.println("getting marks for "+studentname);
11         if(studentname.equals("johny")) {
12             //when we have return inside the catch block.
13             //second return is unreachable.
14             try {
15                 int i=9/0;
16                 return 234234;
17             } catch (ArithmaticException e) {
18                 System.out.println("arithmetic exception .....");
19                 return 23434;
20             }
21             finally {
22                 System.out.println("finally block.....");
23             }
24             return 100;//Unreachable code
25         }else if(studentname.equals("rose")) {
26             return 90;
27         }else if(studentname.equals("tiger")) {
28             return 80;
29         }else {
30             System.out.println("no student found .....");
31             return -1;
32         }
33
34 //      System.out.println("end of the method");//Unreachable code
35     }
36
37     public static void main(String[] args) {
38
39         int marks=getmarks("johny");
40         System.out.println(marks);
41
42         int marks1=testfinally4.getmarks("rose");
43         System.out.println(marks1);
44
45         System.out.println("end of the program");
46
47     }
48

```

```
46
47      }
48
49 }
50
51 //getting marks for johny
52 //234234
53 //getting marks for rose
54 //90
55 //end of the program
56
57
58
59
60
```

Writable

Smart Insert

paste testfinally5-

```

testfinally5.java X testfinally6.java testfinally7.java testfinally8.java testfinally9.java
1 package com.day24;
2
3 public class testfinally5 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10        System.out.println("getting marks for "+studentname);
11        if(studentname.equals("johny")) {
12            //see the flow when catch has return.
13            //first goes to the finally if present and then comes back to catch.
14            //then runs all lines outside try catch finally.
15            try {
16                int i=9/0;
17                return 234234;
18            } catch (ArithmaticException e) {
19                System.out.println("arithmetic exception .....");
20                return 23434;
21            }
22            finally {
23                System.out.println("finally block.....");
24            }
25        //    return 100;//Unreachable code
26        }else if(studentname.equals("rose")) {
27            return 90;
28        }else if(studentname.equals("tiger")) {
29            return 80;
30
31        }else if(studentname.equals("tiger")) {
32            return 80;
33        }else {
34            System.out.println("no student found .....");
35        }
36    }
37
38    public static void main(String[] args) {
39
40        int marks=getmarks("johny");
41        System.out.println(marks);
42
43        int marks1=testfinally5.getmarks("rose");
44        System.out.println(marks1);
45
46        System.out.println("end of the program");
47
48    }
49

```

```

47
48      }
49
50  }
51
52 //getting marks for johny
53 //arithmetic exception .....
54 //finally block.....
55 //23434
56 //getting marks for rose
57 //90
58 //end of the program
59

```

## paste testfinally6-

```

1 package com.day24;
2
3 public class testfinally6 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         System.out.println("getting marks for "+studentname);
11         if(studentname.equals("johny")) {
12             //first finally runs then catch return runs.
13             //then rest of code runs.
14             try {
15                 int i=9/0;
16             } catch (ArithmaticException e) {
17                 System.out.println("arithmetic exception .....");
18                 return 23434;
19             }
20             finally {
21                 System.out.println("finally block.....");
22             }
23             return 100;
24         }else if(studentname.equals("rose")) {
25             return 90;
26         }else if(studentname.equals("tiger")) {
27             return 80;
28         }else {
29             System.out.println("else.....");
30         }
31     }
32 }

```

```
27         return 80;
28     }else {
29         System.out.println("no student found .....");
30         return -1;
31     }
32
33 //      System.out.println("end of the method");//Unreachable code
34 }
35
36@ public static void main(String[] args) {
37
38     int marks=getmarks("johny");
39     System.out.println(marks);
40
41     int marks1=testfinally6.getmarks("rose");
42     System.out.println(marks1);
43
44     System.out.println("end of the program");
45
46 }
47
48 }
```

```
47
48 }
49
50 //getting marks for johny
51 //arithmetic exception .....
52 //finally block.....
53 //23434
54 //getting marks for rose
55 //90
56 //end of the program
57
58
59
60
61
62
63
```

Writable

Smart Insert

1:1:0

paste testfinally7-

```
testfinally7.java testfinally8.java testfinally9.java
1 package com.day24;
2
3 public class testfinally7 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         System.out.println("getting marks for "+studentname);
11         if(studentname.equals("johny")) {
12             //finally with return.
13             try {
14                 int i=9/0;
15             } catch (ArithmaticException e) {
16                 System.out.println("arithmetic exception .....");
17                 return 23434;
18             }
19             //warning when we add return to finally.
20             //finally block does not complete normally
21             //catch return will run.
22             //but finally return will update the return value from catch.
23             finally {
24                 System.out.println("finally block.....");
25                 return 67890;
26             }
27         //         return 100;//Unreachable code
28     }
29 }
```

```
26         }
27     //      return 100;//Unreachable code
28     }else if(studentname.equals("rose")) {
29         return 90;
30     }else if(studentname.equals("tiger")) {
31         return 80;
32     }else {
33         System.out.println("no student found ..... ");
34         return -1;
35     }
36
37 //      System.out.println("end of the method");//Unreachable code
38 }
39
40 public static void main(String[] args) {
41
42     int marks=getmarks("johny");
43     System.out.println(marks);
44
45     int marks1=testfinally7.getmarks("rose");
46     System.out.println(marks1);
47
48     System.out.println("end of the program");
49 }
```

```

47
48     System.out.println("end of the program");
49
50 }
51
52 }
53
54 //getting marks for johny
55 //arithmetic exception .....
56 //finally block.....
57 //67890
58 //getting marks for rose
59 //90
60 //end of the program
61
62

```

## paste testfinally8-

```

1 package com.day24;
2
3 public class testfinally8 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         System.out.println("getting marks for "+studentname);
11         if(studentname.equals("johny")) {
12             //no exception anywhere.
13             try {
14                 int i=9/3;
15                 return 10;
16             } catch (ArithmaticException e) {
17                 System.out.println("arithmetic exception .....");
18                 return 23434;
19             }
20         finally {
21             System.out.println("finally block.....");
22             return 67890;
23         }
24         //         return 100;//Unreachable code
25         }else if(studentname.equals("rose")) {
26             return 90;
27         }else if(studentname.equals("tiger")) {
28             return 80;
29         }else {

```

```
27 }else if(studentname.equals("tiger")) {  
28     return 80;  
29 }else {  
30     System.out.println("no student found .....");  
31     return -1;  
32 }  
33  
34 //    System.out.println("end of the method");//Unreachable code  
35 }  
36  
37 public static void main(String[] args) {  
38  
39     int marks=getmarks("johny");  
40     System.out.println(marks);  
41  
42     int marks1=testfinally8.getmarks("rose");  
43     System.out.println(marks1);  
44  
45     System.out.println("end of the program");  
46  
47 }
```

```
46  
47     }  
48  
49 }  
50  
51 //getting marks for johny  
52 //finally block.....  
53 //67890  
54 //getting marks for rose  
55 //90  
56 //end of the program  
57  
58
```

paste testfinally9-

```

1 package com.day24;
2
3 public class testfinally9 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         System.out.println("getting marks for "+studentname);
11         if(studentname.equals("johny")) {
12             //no return in finally.
13             //no exception.
14             //return statement runs after finally block only.
15             try {
16                 int i=9/3;
17                 return 10;
18             } catch (ArithmaticException e) {
19                 System.out.println("arithmetic exception .....");
20                 return 23434;
21             }
22         finally {
23             System.out.println("finally block.....");
24         }
25         // return 100;//Unreachable code
26         }else if(studentname.equals("rose")) {
27             return 90;
28         }else if(studentname.equals("tiger")) {
29
30             return 90;
31         }else if(studentname.equals("tiger")) {
32             return 80;
33         }else {
34             System.out.println("no student found .....");
35             return -1;
36         }
37
38     public static void main(String[] args) {
39
40         int marks=getmarks("johny");
41         System.out.println(marks);
42
43         int marks1=testfinally9.getmarks("rose");
44         System.out.println(marks1);
45
46         System.out.println("end of the program");
47
}
}

```

```

45
46     System.out.println("end of the program");
47
48 }
49
50 }
51
52 //getting marks for johny
53 //finally block.....
54 //10
55 //getting marks for rose
56 //90
57 //end of the program
58
59

```

## Return statement in catch-

```

7     System.out.println("getting marks for : " + studentName);
8
9     if (studentName.equals("vibha")) {
10         try {
11             int i = 9 / 0;
12         } catch (ArithmaticException e) {
13             System.out.println("AE is coming..");
14             return 10;
15         }
16         return 100;
17     }

```



Return is last statement in the block to run.

No exception-

```

5  public static int getMarks(String studentName) {
6
7      System.out.println("getting marks for : " + studentName);
8
9      if (studentName.equals("vibha")) {
10         try {
11             int i = 9 / 3;
12         } catch (ArithmaticException e) {
13             System.out.println("AE is coming..");
14             return 10;
15         }
16         return 100;
17     }
18
19 }

```



## Finally with return-

```

5  public static int getMarks(String studentName) {
6
7      System.out.println("getting marks for : " + studentName);
8
9      if (studentName.equals("vibha")) {
10         try {
11             int i = 9 / 0;
12         } catch (ArithmaticException e) {
13             System.out.println("AE is coming..");
14             return 10;
15         }
16         finally {
17             System.out.println("bye!!");
18             return 80;
19         }
20     }

```

```

Session/src/ExceptionHandling/TestingFinally.java - Eclipse IDE
Console Problems Javadoc Declaration Resources
<terminated> TestingFinally [Java Application] /Users/naveenautomatio
getting marks for : vibha
AE is coming..
bye!!
80
Name

```

Finally runs even after return statement. see above example, though catch has return but still finally runs.

Without exception-

```

5  public static int getMarks(String studentName) {
6
7      System.out.println("getting marks for : " + studentName);
8
9      if (studentName.equals("vibha")) {
10          try {
11              int i = 9 / 3;
12          }
13          catch (ArithmaticException e) {
14              System.out.println("AE is coming..");
15              return 10;
16          }
17          finally {
18              System.out.println("bye!!");
19              return 80;
20          }
21      }
22

```

```
aSessions/src/ExceptionHandling/TestingFinally.java - Eclipse IDE


```

## Some more code changes-

```
5  public static int getMarks(String studentName) {
6
7      System.out.println("getting marks for : " + studentName);
8
9      if (studentName.equals("vibha")) {
10         try {
11             int i = 9 / 3;
12             return 30;
13         }
14         catch (ArithmaticException e) {
15             System.out.println("AE is coming..");
16             return 10;
17         }
18         finally {
19             System.out.println("bye!!!");
20             return 80;
21         }
22     }
```

```
vaSessions/src/ExceptionHandling/TestingFinally.java - Eclipse IDE


```

## Some more code changes-

```

4
5  public static int getMarks(String studentName) {
6
7      System.out.println("getting marks for : " + studentName);
8
9      if (studentName.equals("vibha")) {
10         try {
11             int i = 9 / 3;
12             return 30;
13         }
14         catch (ArithmaticException e) {
15             System.out.println("AE is coming..");
16             return 10;
17         }
18         finally {
19             System.out.println("bye!!");
20             //return 80;
21         }
22     }
23 }

```

Sessions/src/ExceptionHandling/TestingFinally.java - Eclipse

```

Console Problems Javadoc Declarations
<terminated> TestingFinally [Java Application] /Users/naveen
getting marks for : vibha
bye!!
30

```

## Unreachable return-

```

5  public static int getMarks(String studentName) {
6
7      System.out.println("getting marks for : " + studentName);
8
9      if (studentName.equals("vibha")) {
10         try {
11             int i = 9 / 3;
12             return 30;
13         }
14         catch (ArithmaticException e) {
15             System.out.println("AE is coming..");
16             return 10;
17         }
18         finally {
19             System.out.println("bye!!");
20             //return 80;
21         }
22         return 50;    }
23 }

```

## System exit –

```

3 public class TestingFinally {
4
5     public static int getMarks(String studentName) {
6
7         System.out.println("getting marks for : " + studentName);
8
9         if (studentName.equals("vibha")) {
10            try {
11                int i = 9 / 3;
12                System.exit(1); //shutdown the JVM
13                return 30;
14            }
15            catch (ArithmeticException e) {
16                System.out.println("AE is coming..");
17                return 10;
18            }
19            finally {
20                System.out.println("bye!!");
21                return 80;
22            }
23        }
24    }

```

It will terminate and no more statements run.

paste testfinally10-

```

1 package com.day24;
2
3 public class testfinally10 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10        //exit zero.
11        System.out.println("getting marks for "+studentname);
12        if(studentname.equals("johny")) {
13            try {
14                int i=9/3;
15                System.exit(0); //shutdown jvm.
16                return 10;
17            } catch (ArithmeticException e) {
18                System.out.println("arithmetic exception .....");
19                return 23434;
20            }
21            finally {
22                System.out.println("finally block.....");
23            }
24        //      return 100;//Unreachable code
25        }else if(studentname.equals("rose")) {
26            return 90;
27        }else if(studentname.equals("tiger")) {
28            return 80;
29        }else {

```

```
27     }else if(studentname.equals("tiger")) {  
28         return 80;  
29     }else {  
30         System.out.println("no student found .....");  
31         return -1;  
32     }  
33  
34 //    System.out.println("end of the method");//Unreachable code  
35 }  
36  
37 public static void main(String[] args) {  
38  
39     int marks=getmarks("johny");  
40     System.out.println(marks);  
41  
42     int marks1=testfinally10.getmarks("rose");  
43     System.out.println(marks1);  
44  
45     System.out.println("end of the program");  
46 }
```

```
44     System.out.println("end of the program");  
45  
46 }  
47 }  
48  
49 }  
50  
51 //getting marks for johny  
52  
53 }
```

paste test finally11-



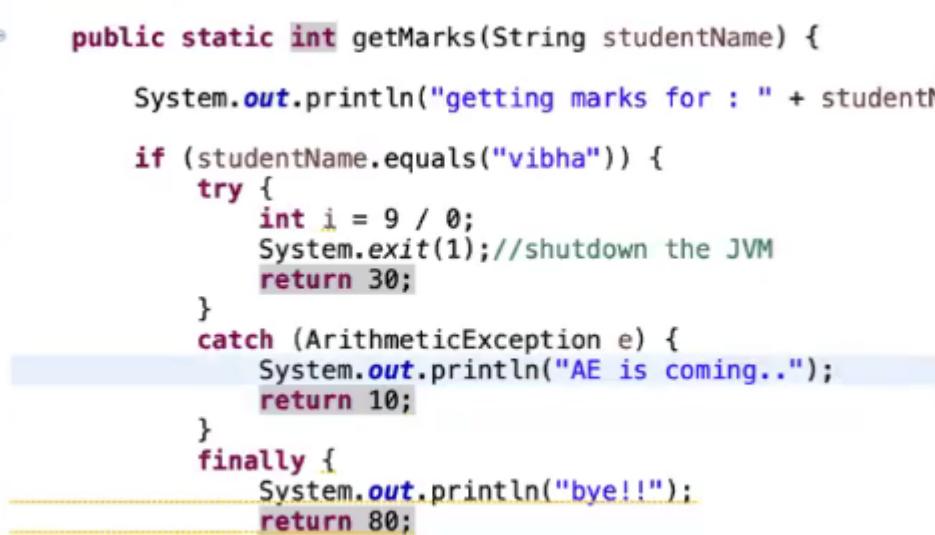
```

1 package com.day24;
2
3 public class testfinally11 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         //exit one.
11         //zero and one works same way.
12         System.out.println("getting marks for "+studentname);
13         if(studentname.equals("johny")) {
14             try {
15                 int i=9/3;
16                 System.exit(1); //shutdown jvm.
17                 return 10;
18             } catch (ArithmaticException e) {
19                 System.out.println("arithmetic exception .....");
20                 return 23434;
21             }
22             finally {
23                 System.out.println("finally block.....");
24             }
25         }
26         //Unreachable code
27         }else if(studentname.equals("rose")) {
28             return 90;
29         }else if(studentname.equals("tiger")) {
30             return 80;
31         }else {
32             System.out.println("no student found .....");
33             return -1;
34         }
35         //Unreachable code
36     }
37
38     public static void main(String[] args) {
39
40         int marks=getmarks("johny");
41         System.out.println(marks);
42
43         int marks1=testfinally11.getmarks("rose");
44         System.out.println(marks1);
45
46         System.out.println("end of the program");
47

```

```
45
46     System.out.println("end of the program");
47
48 }
49
50 }
51
52 //getting marks for johny
53
```

## When will exception block come with exit-



```
FinallyBlock.java  TestingFinally.java
1  public class TestingFinally {
2
3      public static int getMarks(String studentName) {
4
5          System.out.println("getting marks for : " + studentName);
6
7          if (studentName.equals("vibha")) {
8              try {
9                  int i = 9 / 0;
10                 System.exit(1); //shutdown the JVM
11                 return 30;
12             }
13             catch (ArithmaticException e) {
14                 System.out.println("AE is coming..");
15                 return 10;
16             }
17             finally {
18                 System.out.println("bye!!!");
19                 return 80;
20             }
21         }
22     }
23 }
```

```
SessionHandling/TestingFinally.java - Eclipse IDE
+---> Console Problems Javadoc Declaration Help
<terminated> TestingFinally [Java Application] /Users/naveenautomat
getting marks for : vibha
AE is coming..
bye!!
80
```

## paste testfinally12-

```

1 package com.day24;
2
3 public class testfinally12 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         //exit zero.
11         //when exception, exit statement is not reached and programs as usual.
12         //here return of catch is taken after finally runs.
13         System.out.println("getting marks for "+studentname);
14         if(studentname.equals("johny")) {
15             try {
16                 int i=9/0;
17                 System.exit(0); //shutdown jvm.
18                 return 10;
19             } catch (ArithmaticException e) {
20                 System.out.println("arithmetic exception .....");
21                 return 23434;
22             }
23             finally {
24                 System.out.println("finally block.....");
25             }
26         }
27         return 100;//Unreachable code
28     } else if(studentname.equals("rose")) {
29         return 90;
30     } else if(studentname.equals("tiger")) {
31         return 90;
32     } else {
33         System.out.println("no student found .....");
34         return -1;
35     }
36     // System.out.println("end of the method");//Unreachable code
37 }
38
39     public static void main(String[] args) {
40
41         int marks=getmarks("johny");
42         System.out.println(marks);
43
44         int marks1=testfinally12.getmarks("rose");
45         System.out.println(marks1);
46
47         System.out.println("end of the program");
48

```

```

46
47     System.out.println("end of the program");
48
49 }
50
51 }
52
53 //getting marks for johny
54 //arithmetic exception .....
55 //finally block.....
56 //23434
57 //getting marks for rose
58 //90
59 //end of the program
60

```

## paste testfinally13-

```

1 package com.day24;
2
3 public class testfinally13 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10         //exit zero.
11         //when exception, exit statement is not reached and programs as usual.
12         //here return of finally taken.
13         System.out.println("getting marks for "+studentname);
14         if(studentname.equals("johny")) {
15             try {
16                 int i=9/0;
17                 System.exit(0); //shutdown jvm.
18                 return 10;
19             } catch (ArithmaticException e) {
20                 System.out.println("arithmetic exception .....");
21                 return 23434;
22             }
23             finally {
24                 System.out.println("finally block.....");
25                 return 10;
26             }
27         }
28         //return 100;//Unreachable code
29         else if(studentname.equals("rose")) {
30             return 90;
31         }
32     }
33 }

```

```
28          }  
29          }  
30          }else if(studentname.equals("rose")) {  
31              return 90;  
32          }else if(studentname.equals("tiger")) {  
33              return 80;  
34          }else {  
35              System.out.println("no student found .....");  
36          }  
37      }  
38  }  
39  
40  public static void main(String[] args) {  
41  
42      int marks=getmarks("johny");  
43      System.out.println(marks);  
44  
45      int marks1=testfinally13.getmarks("rose");  
46      System.out.println(marks1);  
47  
48      System.out.println("end of the program");  
49  }  
50 }  
51  
52 }  
53  
54 //getting marks for johny  
55 //arithmetic exception .....  
56 //finally block.....  
57 //10  
58 //getting marks for rose  
59 //90  
60 //end of the program  
61  
62 }
```

```
49  
50 }  
51  
52 }  
53  
54 //getting marks for johny  
55 //arithmetic exception .....  
56 //finally block.....  
57 //10  
58 //getting marks for rose  
59 //90  
60 //end of the program  
61  
62 }
```

paste testfinally14-

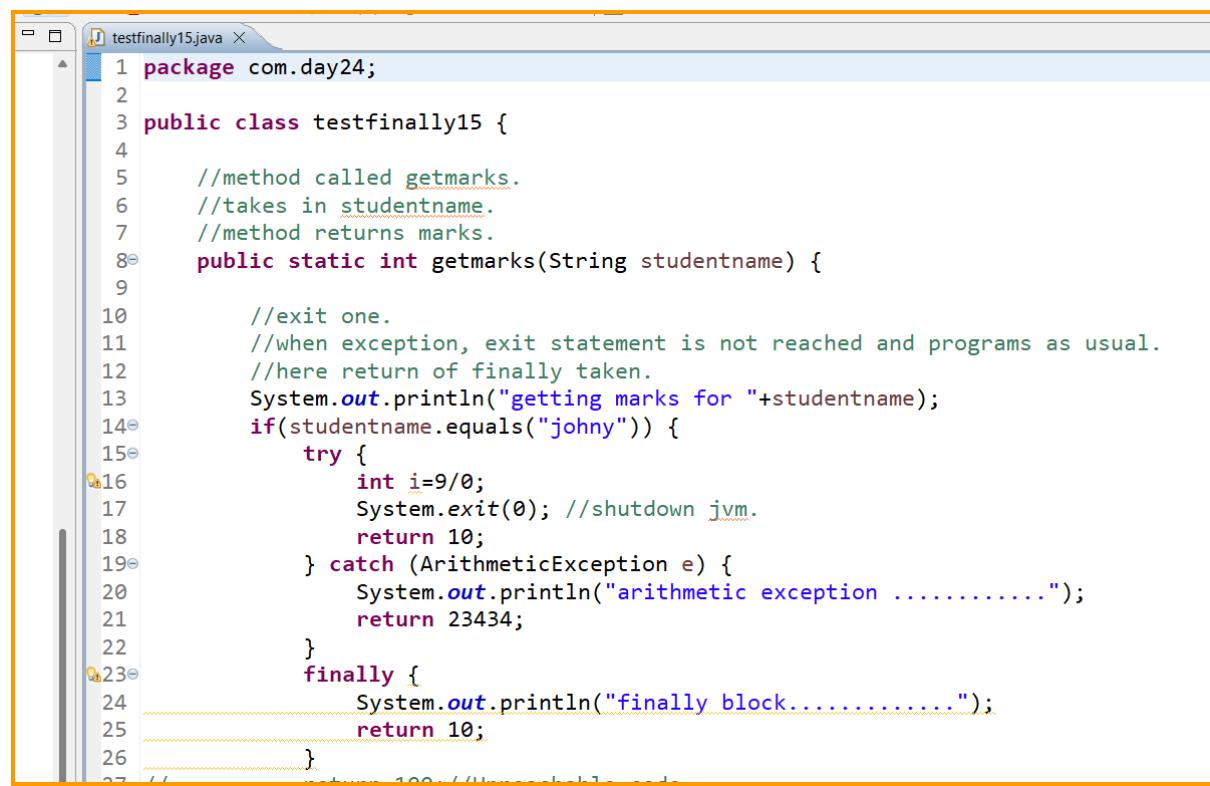
```
1 package com.day24;
2
3 public class testfinally14 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10        //exit one.
11        //when exception, exit statement is not reached and programs as usual.
12        //here return of catch is taken after finally runs.
13        System.out.println("getting marks for "+studentname);
14        if(studentname.equals("johny")) {
15            try {
16                int i=9/0;
17                System.exit(0); //shutdown jvm.
18                return 10;
19            } catch (ArithmaticException e) {
20                System.out.println("arithmetic exception .....");
21                return 23434;
22            }
23            finally {
24                System.out.println("finally block.....");
25            }
26        }
27        return 100;//Unreachable code
28    }else if(studentname.equals("rose")) {
29        return 90;
30    }else if(studentname.equals("tiger")) {
31
32        }else if(studentname.equals("rose")) {
33            return 90;
34        }else if(studentname.equals("tiger")) {
35            return 80;
36        }else {
37            System.out.println("no student found .....");
38            return -1;
39        }
40
41        // System.out.println("end of the method");//Unreachable code
42    }
43
44    public static void main(String[] args) {
45
46        int marks=getmarks("johny");
47        System.out.println(marks);
48
49        int marks1=testfinally14.getmarks("rose");
50        System.out.println(marks1);
51
52        System.out.println("end of the program");
53    }
54}
```

```

46
47         System.out.println("end of the program");
48
49     }
50
51 }
52
53 //getting marks for johny
54 //arithmetic exception .....
55 //finally block.....
56 //23434
57 //getting marks for rose
58 //90
59 //end of the program
60

```

## paste testfinally15-



```

1 package com.day24;
2
3 public class testfinally15 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9
10        //exit one.
11        //when exception, exit statement is not reached and programs as usual.
12        //here return of finally taken.
13        System.out.println("getting marks for "+studentname);
14        if(studentname.equals("johny")) {
15            try {
16                int i=9/0;
17                System.exit(0); //shutdown jvm.
18                return 10;
19            } catch (ArithmaticException e) {
20                System.out.println("arithmetic exception ....");
21                return 23434;
22            }
23        finally {
24            System.out.println("finally block.....");
25            return 10;
26        }
27    }

```

```

25         return 10;
26     }
27 //      return 100;//Unreachable code
28 }else if(studentname.equals("rose")) {
29     return 90;
30 }else if(studentname.equals("tiger")) {
31     return 80;
32 }else {
33     System.out.println("no student found .....");
34     return -1;
35 }
36
37 //      System.out.println("end of the method");//Unreachable code
38 }
39
40 public static void main(String[] args) {
41
42     int marks=getmarks("johny");
43     System.out.println(marks);
44
45     int marks1=testfinally15.getmarks("rose");
46     System.out.println(marks1);
47
48     System.out.println("end of the program");
49
50 }
51

```

```

49
50 }
51
52 }
53
54 //getting marks for johny
55 //arithmetic exception .....
56 //finally block.....
57 //10
58 //getting marks for rose
59 //90
60 //end of the program
61

```

Try finally without catch-

```

21      }
22
23      else if (studentName.equals("monika")) {
24          return 90;
25      }
26
27      else if (studentName.equals("veena")) {
28          return 40;
29      }
30
31      else {
32          System.out.println("no student found...");
33
34          try {
35              int p = 9 / 0;
36          } finally {
37              return 5;
38          }
39
40      }
41
42
43  public static void main(String[] args) {
44
45      int m1 = TestingFinally.getMarks("naveen");
46      System.out.println(m1);
47
48  }
49

```

The screenshot shows the Eclipse IDE interface. The title bar says '124JavaSessions/src/ErrorHandling/TestingFinally.java - Eclipse'. Below the title bar is a toolbar with various icons. The main area shows the Java code for 'TestingFinally.java'. In the bottom right corner of the code editor, there is a small yellow status bar with the text '5'. To the right of the code editor is the 'Console' tab, which displays the application's output:

```

<terminated> TestingFinally [Java Application] /Users/naveenau
getting marks for : naveen
no student found...
5

```

paste testfinally16-

```

testfinally16.java X
1 package com.day24;
2
3 public class testfinally16 {
4
5     //method called getmarks.
6     //takes in studentname.
7     //method returns marks.
8     public static int getmarks(String studentname) {
9         //try finally without catch.
10
11         System.out.println("getting marks for "+studentname);
12         if(studentname.equals("johny")) {
13             return 100;
14         }else if(studentname.equals("rose")) {
15             return 90;
16         }else if(studentname.equals("tiger")) {
17             return 80;
18         }else {
19             System.out.println("no student found ..... ");
20             try {
21                 int p=9/0;
22             }finally {//finally block does not complete normally
23                 //warning message.
24                 return 5;
25             }
26         }
27     }
28     //      System.out.println("end of the method");//Unreachable code
29 }
30
31     public static void main(String[] args) {
32
33         int marks=getmarks("karan");
34         System.out.println(marks);
35
36         int marks1=testfinally16.getmarks("rose");
37         System.out.println(marks1);
38
39         System.out.println("end of the program");
40
41     }
42
43 }
44

```

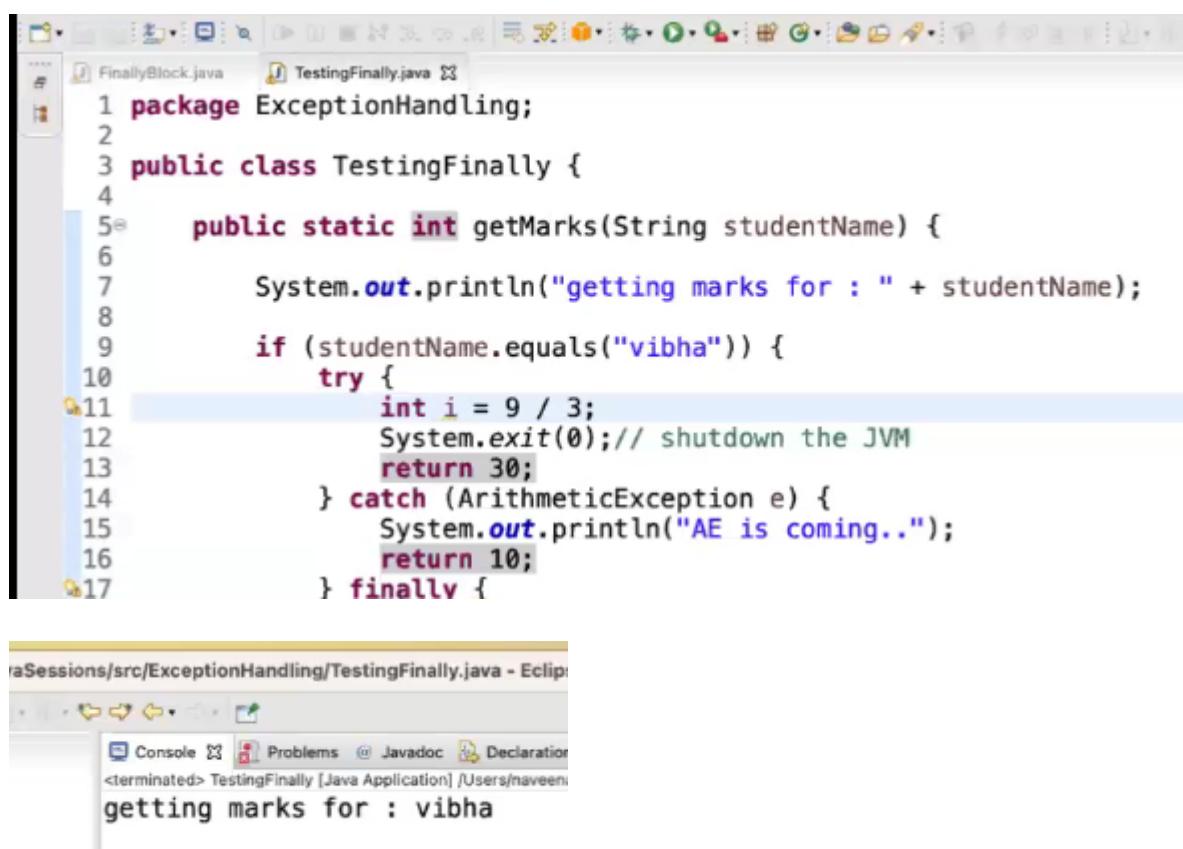
```

42
43 }
44
45 //getting marks for karan
46 //no student found .....
47 //5
48 //getting marks for rose
49 //90
50 //end of the program
51

```

Exit 0-

Partial shutdown.



```

1 package ExceptionHandling;
2
3 public class TestingFinally {
4
5     public static int getMarks(String studentName) {
6
7         System.out.println("getting marks for : " + studentName);
8
9         if (studentName.equals("vibha")) {
10             try {
11                 int i = 9 / 3;
12                 System.exit(0); // shutdown the JVM
13                 return 30;
14             } catch (ArithmeticException e) {
15                 System.out.println("AE is coming..");
16                 return 10;
17             } finally {

```

Output from Console:

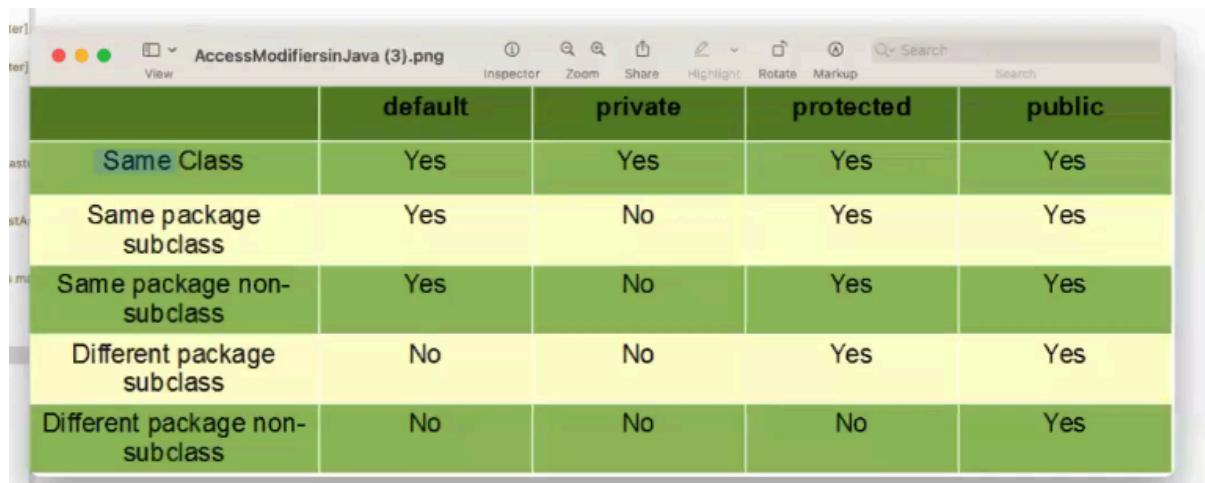
```

getting marks for : vibha

```

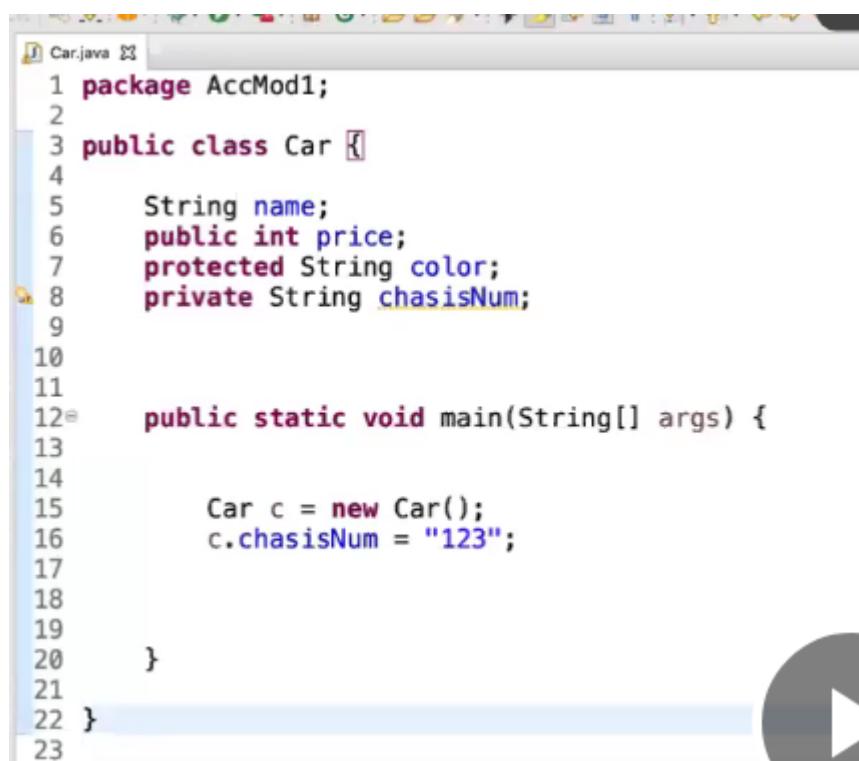
Exit 1 is full shutdown.

Exit 0 is partial shutdown.



The table illustrates the visibility of Java access modifiers (default, private, protected, public) across various class contexts:

	default	private	protected	public
Same Class	Yes	Yes	Yes	Yes
Same package subclass	Yes	No	Yes	Yes
Same package non-subclass	Yes	No	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package non-subclass	No	No	No	Yes

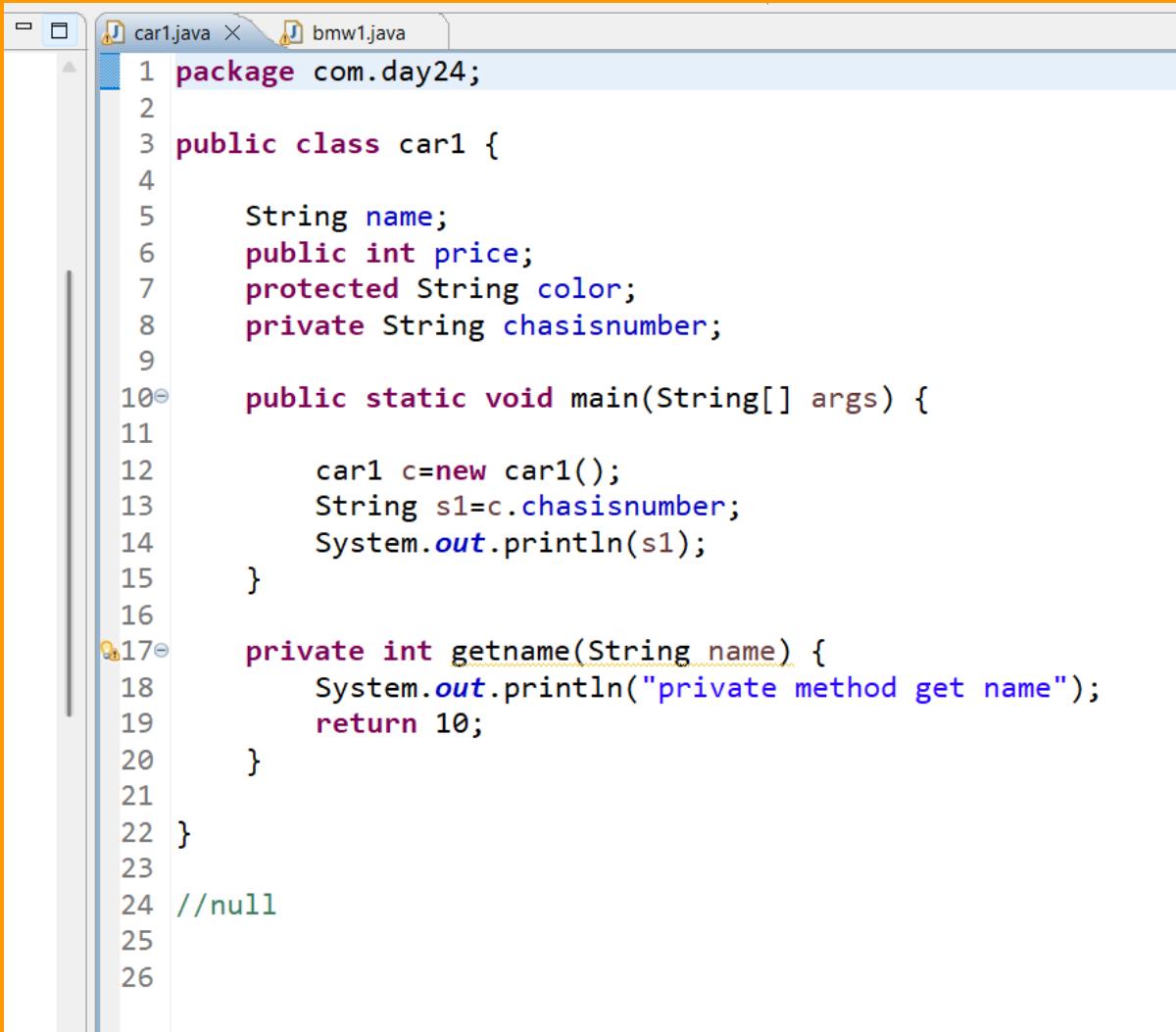


```

1 package AccMod1;
2
3 public class Car {
4
5     String name;
6     public int price;
7     protected String color;
8     private String chasisNum;
9
10
11
12     public static void main(String[] args) {
13
14
15         Car c = new Car();
16         c.chasisNum = "123";
17
18
19     }
20
21
22 }
23

```

paste car1-

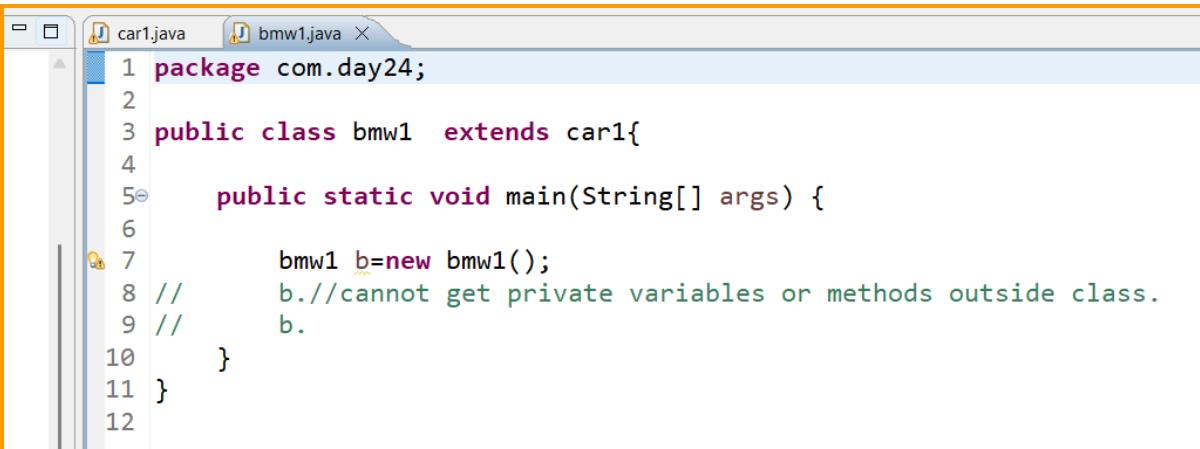


```

1 package com.day24;
2
3 public class car1 {
4
5     String name;
6     public int price;
7     protected String color;
8     private String chasisnumber;
9
10    public static void main(String[] args) {
11
12        car1 c=new car1();
13        String s1=c.chasisnumber;
14        System.out.println(s1);
15    }
16
17    private int getname(String name) {
18        System.out.println("private method get name");
19        return 10;
20    }
21
22 }
23
24 //null
25
26

```

paste bmw1-



```

1 package com.day24;
2
3 public class bmw1  extends car1{
4
5    public static void main(String[] args) {
6
7        bmw1 b=new bmw1();
8        // b.//cannot get private variables or methods outside class.
9        // b.
10    }
11 }
12

```

Second row example-

```

1 package AccMod1;
2
3 public class BMW extends Car{
4
5     public static void main(String[] args) {
6
7
8         BMW b = new BMW();
9         b.ch
10
11
12     }
13
14 }
15
16

```

Third row example-

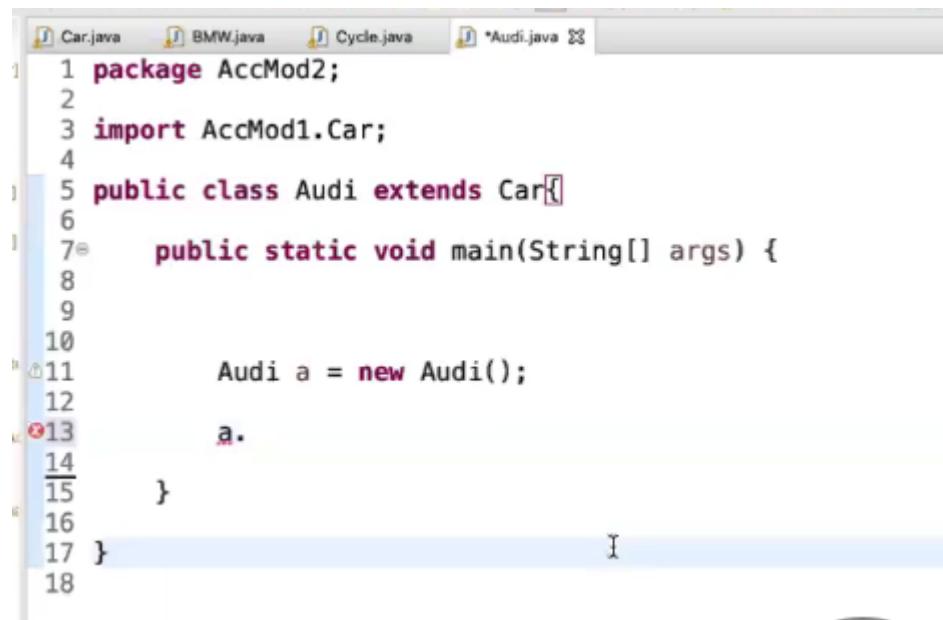
cannot access private variables and methods outside class even object of car created.

```

1 package AccMod1;
2
3 public class Cycle {
4
5     public static void main(String[] args) {
6
7         Car c = new Car();
8         c.
9
10
11     }
12
13 }
14
15

```

Fourth row example-

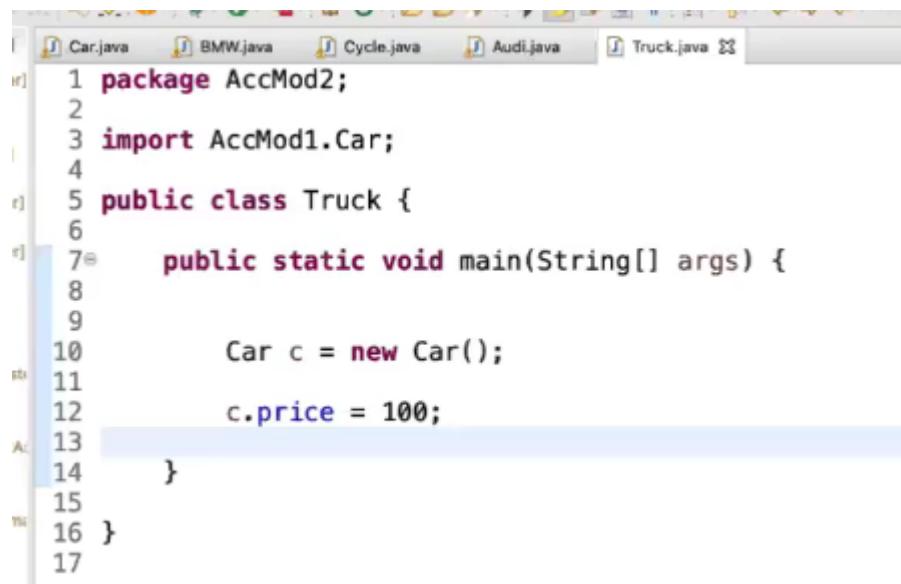


```

1 package AccMod2;
2
3 import AccMod1.Car;
4
5 public class Audi extends Car{
6
7     public static void main(String[] args) {
8
9
10        Audi a = new Audi();
11
12        a.
13
14    }
15
16 }
17
18

```

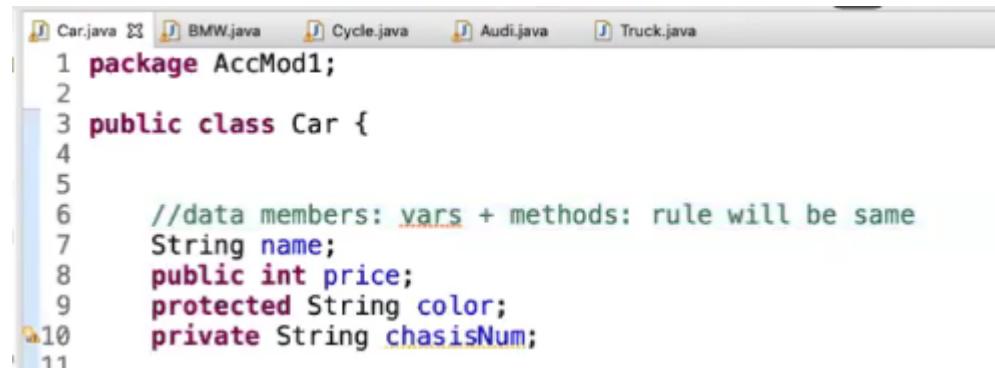
## Fifth row example-



```

1 package AccMod2;
2
3 import AccMod1.Car;
4
5 public class Truck {
6
7     public static void main(String[] args) {
8
9
10        Car c = new Car();
11
12        c.price = 100;
13
14    }
15
16 }
17

```

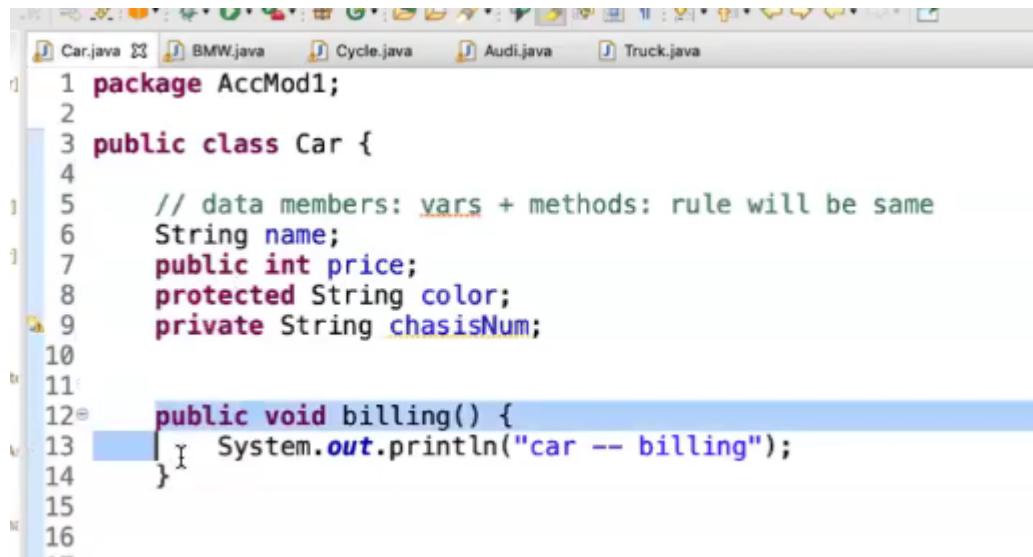


```

1 package AccMod1;
2
3 public class Car {
4
5
6     //data members: vars + methods: rule will be same
7     String name;
8     public int price;
9     protected String color;
10    private String chassisNum;
11

```

## Public methods can be overridden-

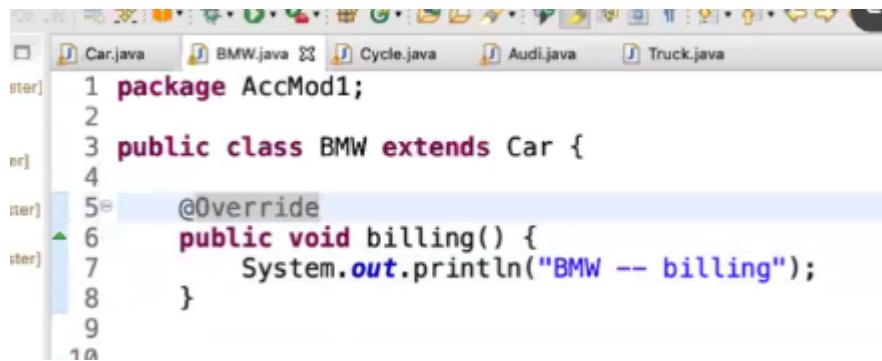


```

1 package AccMod1;
2
3 public class Car {
4
5     // data members: vars + methods: rule will be same
6     String name;
7     public int price;
8     protected String color;
9     private String chasisNum;
10
11
12     public void billing() {
13         System.out.println("car -- billing");
14     }
15
16

```

Bmw-



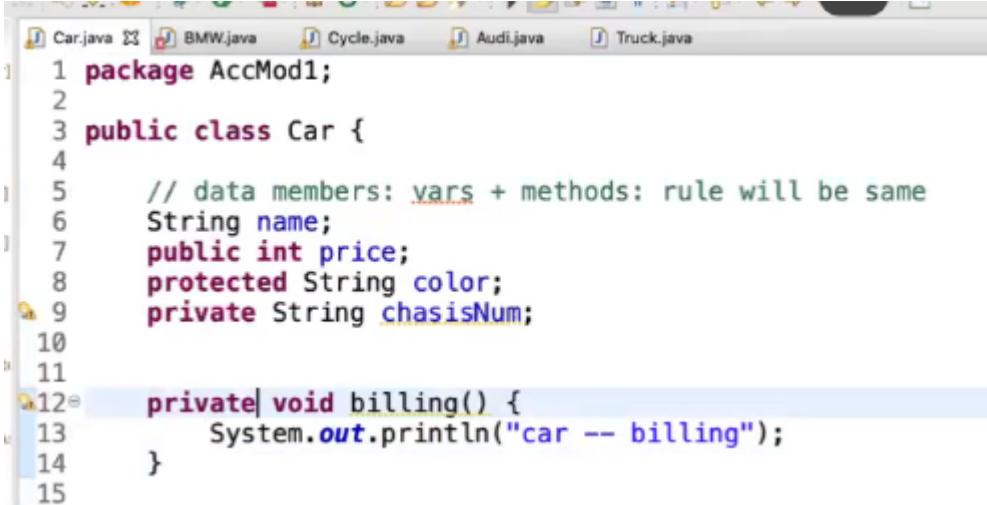
```

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     public void billing() {
7         System.out.println("BMW -- billing");
8     }
9
10

```

Private methods cannot be overridden-

we cant access only outside class.



```

1 package AccMod1;
2
3 public class Car {
4
5     // data members: vars + methods: rule will be same
6     String name;
7     public int price;
8     protected String color;
9     private String chasisNum;
10
11
12    private void billing() {
13        System.out.println("car -- billing");
14    }
15

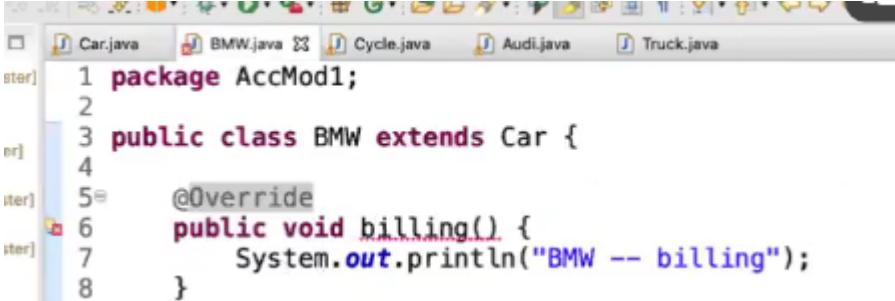
```

## Bmw error-

```

//The method getname(String) of type bmw3
//must override or implement a supertype method

```

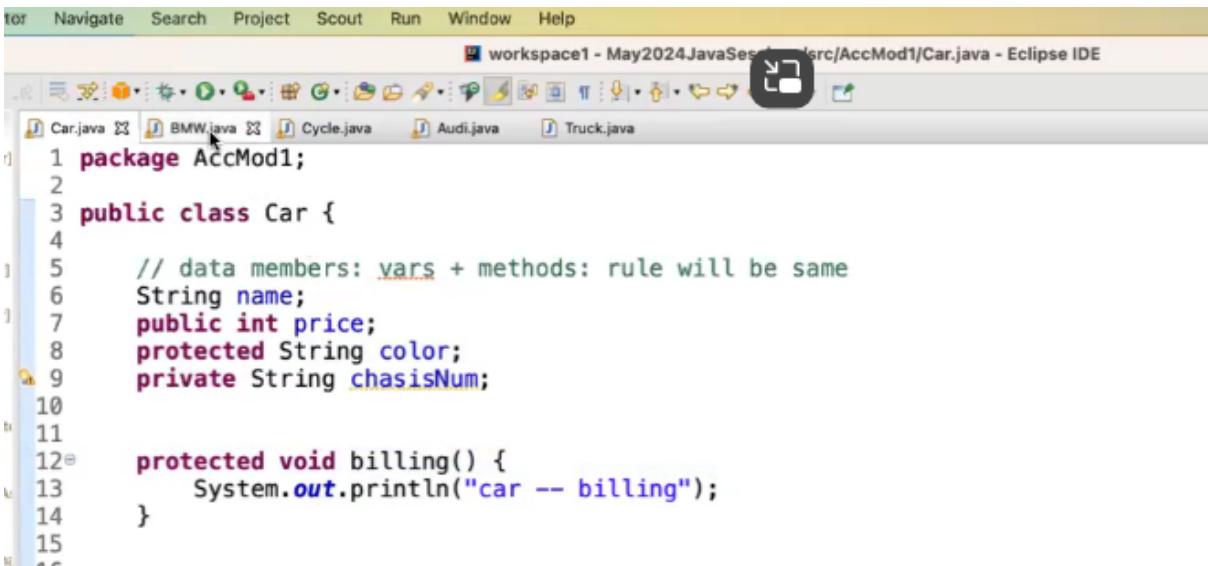


```

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     public void billing() {
7         System.out.println("BMW -- billing");
8     }

```

## Protected-



```

1 package AccMod1;
2
3 public class Car {
4
5     // data members: vars + methods: rule will be same
6     String name;
7     public int price;
8     protected String color;
9     private String chasisNum;
10
11
12    protected void billing() {
13        System.out.println("car -- billing");
14    }
15

```

Always go towards right.

If parent class has protected, then in child class can have public or protected only.

Bmw-

```

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     protected void billing() {
7         System.out.println("BMW -- billing");
8     }
9

```

Private not allowed-

// Cannot reduce the visibility of the inherited method from car2

```

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     private void billing() {
7         System.out.println("BMW -- billing");
8     }
9

```

Default not allowed in bmw-

// Cannot reduce the visibility of the inherited method from car2

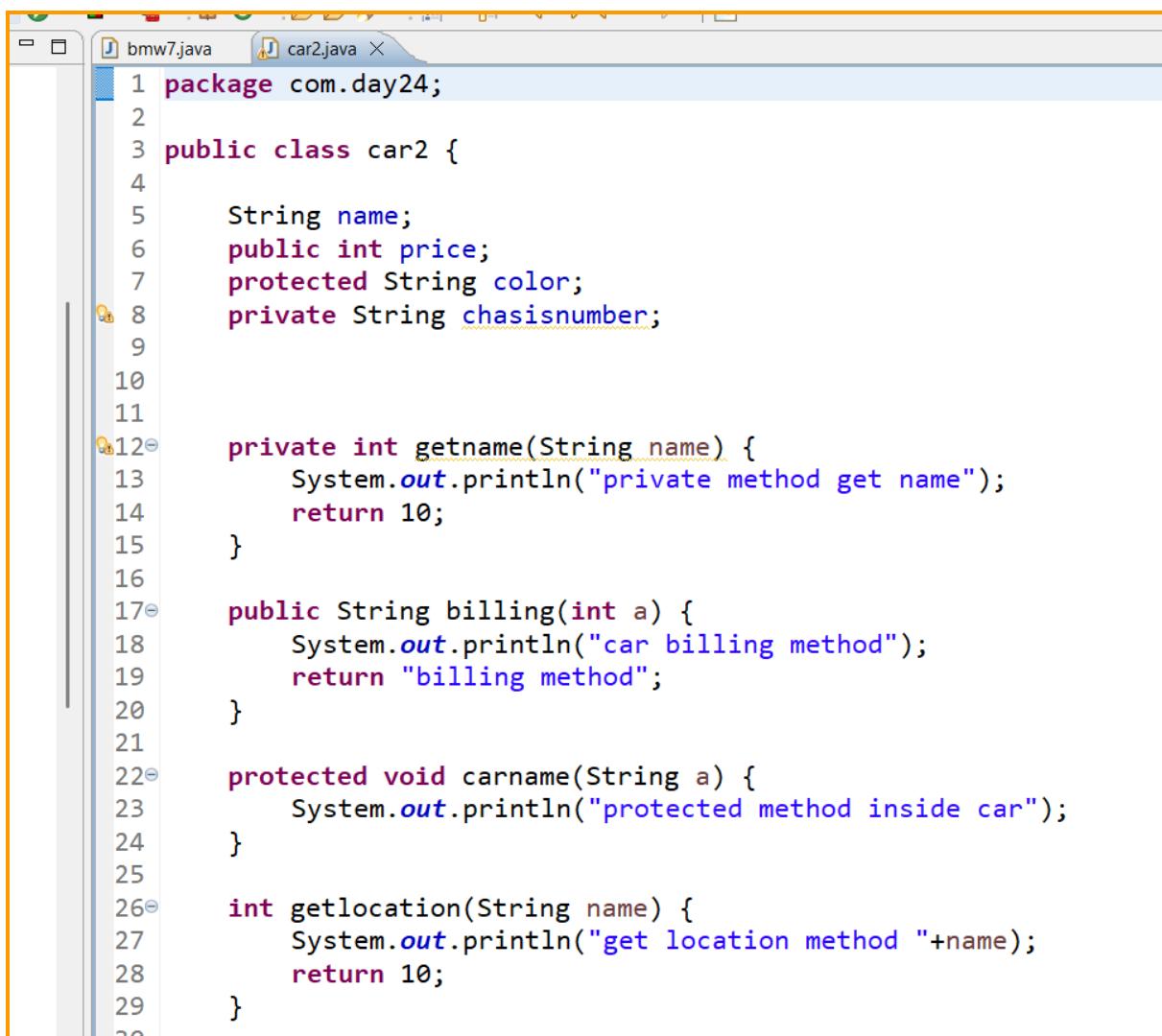
```

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     void billing() {
7         System.out.println("BMW -- billing");
8     }
9

```

protected can become public-

paste car2-

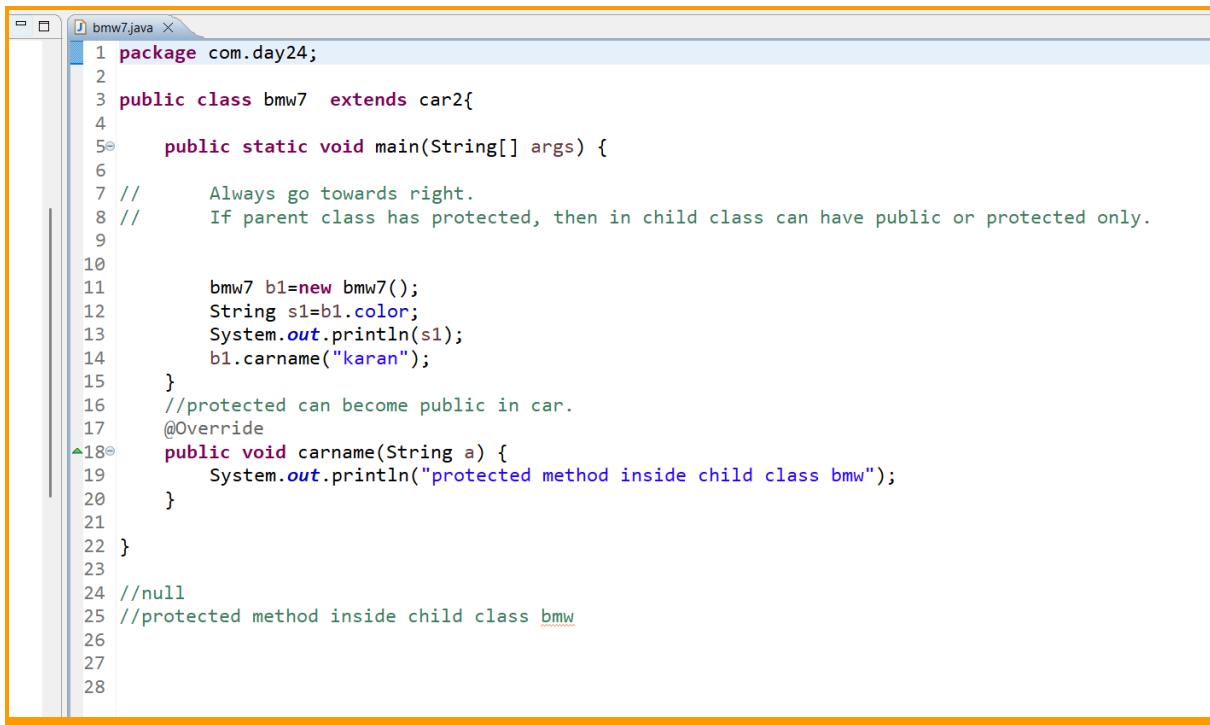


```

1 package com.day24;
2
3 public class car2 {
4
5     String name;
6     public int price;
7     protected String color;
8     private String chasisnumber;
9
10
11
12     private int getname(String name) {
13         System.out.println("private method get name");
14         return 10;
15     }
16
17     public String billing(int a) {
18         System.out.println("car billing method");
19         return "billing method";
20     }
21
22     protected void carname(String a) {
23         System.out.println("protected method inside car");
24     }
25
26     int getlocation(String name) {
27         System.out.println("get location method "+name);
28         return 10;
29     }
30
31 }
32

```

paste bmw7-

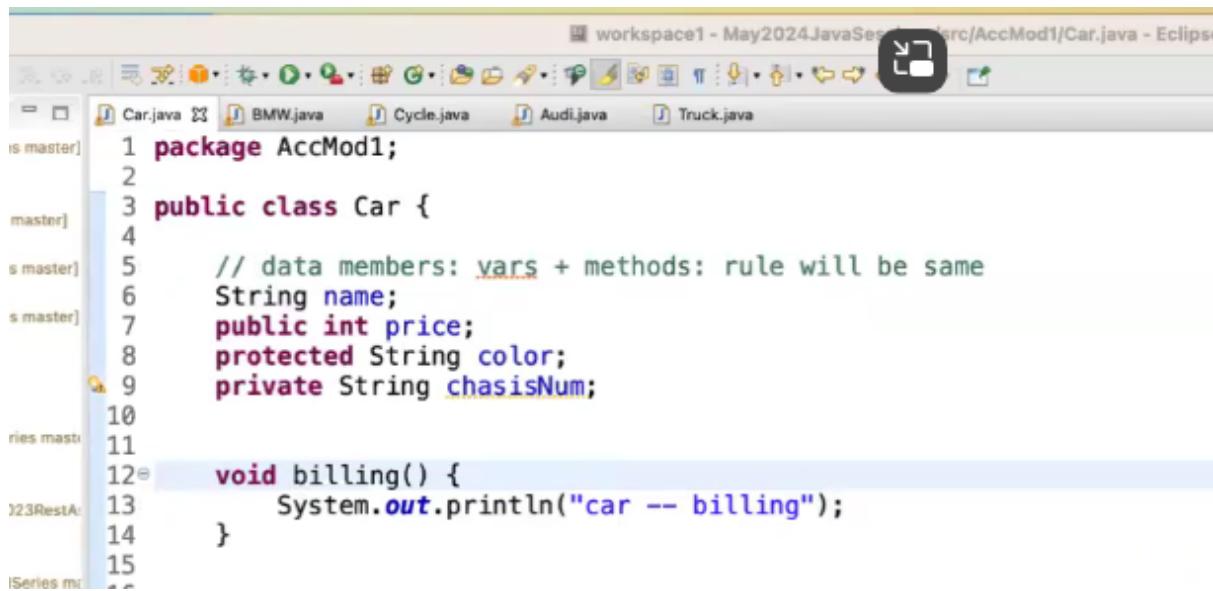


```

1 package com.day24;
2
3 public class bmw7 extends car2{
4
5     public static void main(String[] args) {
6
7         // Always go towards right.
8         // If parent class has protected, then in child class can have public or protected only.
9
10
11        bmw7 b1=new bmw7();
12        String s1=b1.color;
13        System.out.println(s1);
14        b1.carname("karan");
15    }
16    //protected can become public in car.
17    @Override
18    public void carname(String a) {
19        System.out.println("protected method inside child class bmw");
20    }
21
22 }
23
24 //null
25 //protected method inside child class bmw
26
27
28

```

## Default in parent class-

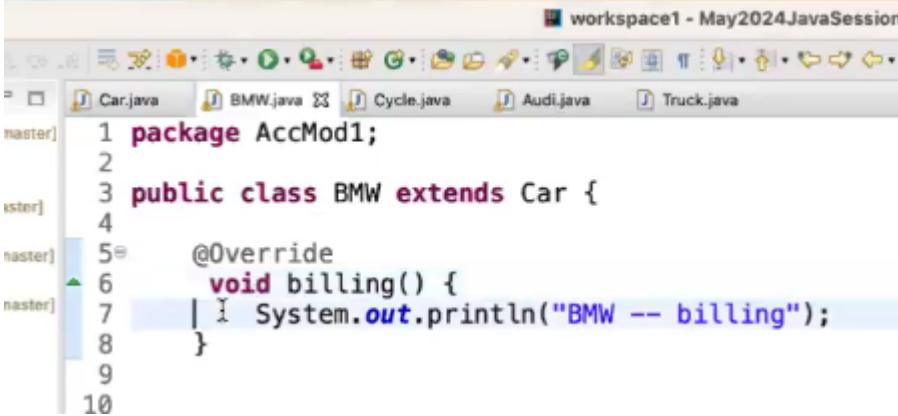


```

1 package AccMod1;
2
3 public class Car {
4
5     // data members: vars + methods: rule will be same
6     String name;
7     public int price;
8     protected String color;
9     private String chassisNum;
10
11
12     void billing() {
13         System.out.println("car -- billing");
14     }
15

```

## Child can have default-



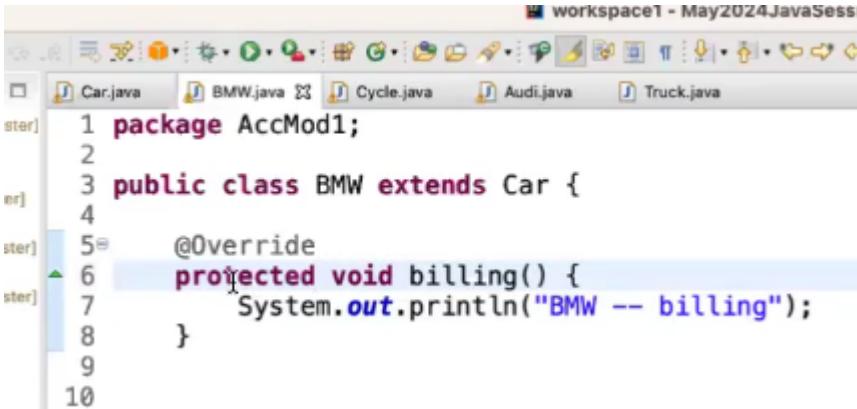
```

workspace1 - May2024JavaSession
Car.java  BMW.java  Cycle.java  Audi.java  Truck.java

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     void billing() {
7         System.out.println("BMW -- billing");
8     }
9
10

```

## Protected allowed-



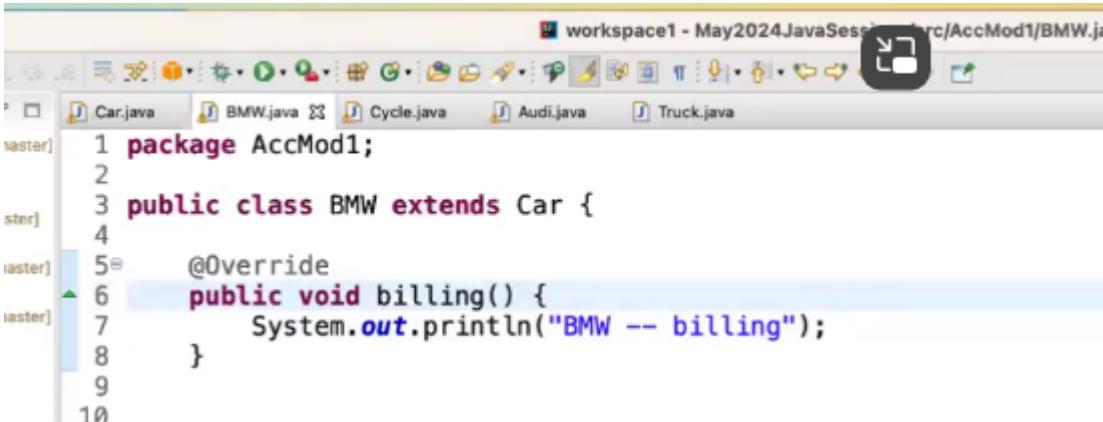
```

workspace1 - May2024JavaSession
Car.java  BMW.java  Cycle.java  Audi.java  Truck.java

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     protected void billing() {
7         System.out.println("BMW -- billing");
8     }
9
10

```

## Public allowed-



```

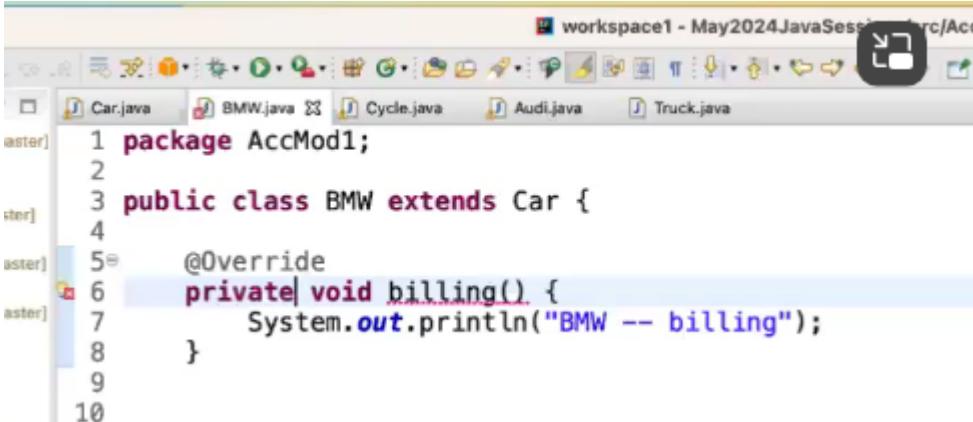
workspace1 - May2024JavaSession
Car.java  BMW.java  Cycle.java  Audi.java  Truck.java

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     public void billing() {
7         System.out.println("BMW -- billing");
8     }
9
10

```

## Private – not allowed-

//Cannot reduce the visibility of the inherited method from car2

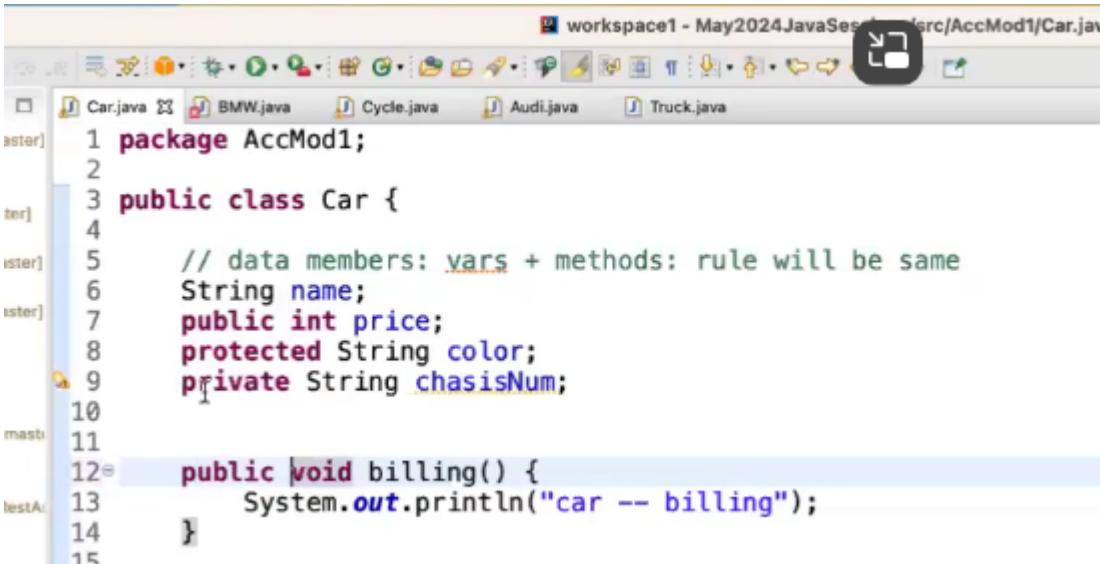


```

workspace1 - May2024JavaSession1 /src/AccMod1/Car.java
1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     private void billing() {
7         System.out.println("BMW -- billing");
8     }
9
10

```

Public in parent-



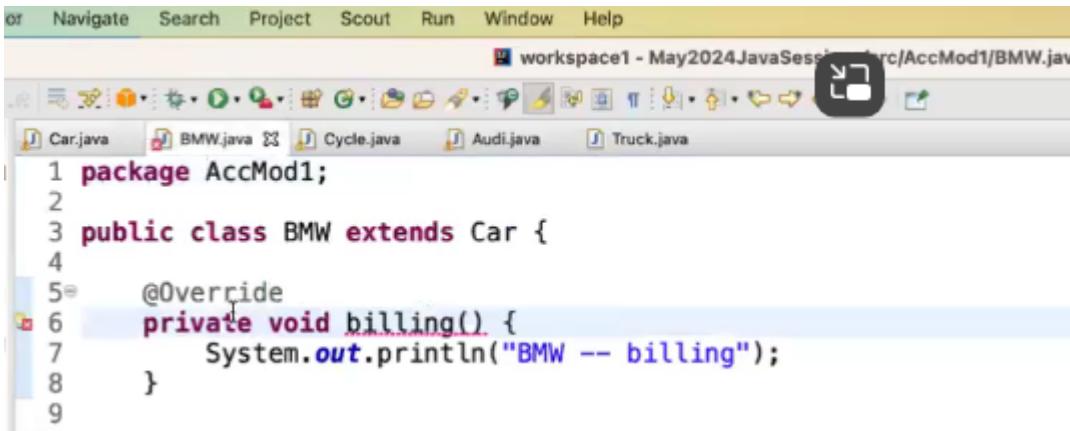
```

workspace1 - May2024JavaSession1 /src/AccMod1/Car.java
1 package AccMod1;
2
3 public class Car {
4
5     // data members: vars + methods: rule will be same
6     String name;
7     public int price;
8     protected String color;
9     private String chasisNum;
10
11
12     public void billing() {
13         System.out.println("car -- billing");
14     }
15

```

Private not allowed in child-

// Cannot reduce the visibility of the inherited method from car2



```

workspace1 - May2024JavaSession1 /src/AccMod1/BMW.java
1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     private void billing() {
7         System.out.println("BMW -- billing");
8     }
9

```

## Default not allowed in child-

//Cannot reduce the visibility of the inherited method from car2

```

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     void billing() {
7         System.out.println("BMW -- billing");
8     }
9
10

```

## Protected not allowed in child-

//Cannot reduce the visibility of the inherited method from car2

```

1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     protected void billing() {
7         System.out.println("BMW -- billing");
8     }
9
10

```

## Public allowed in child-

```

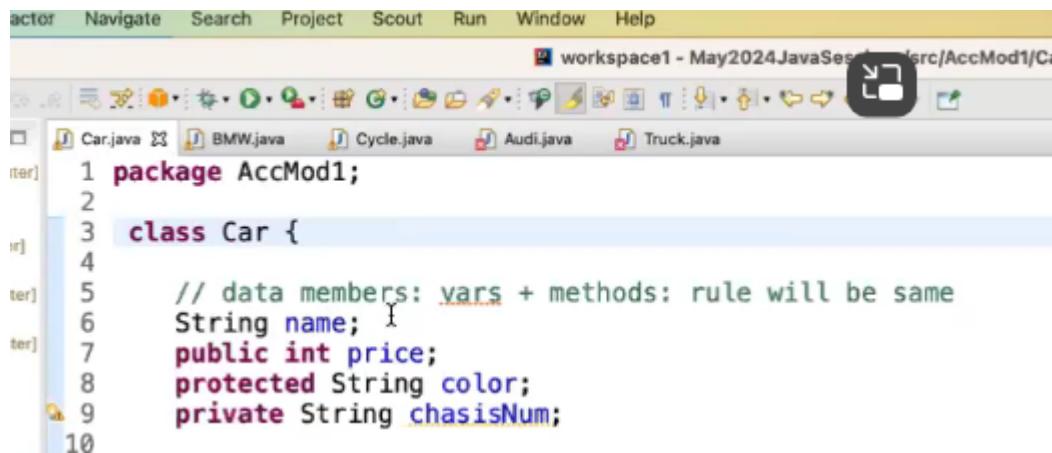
1 package AccMod1;
2
3 public class BMW extends Car {
4
5     @Override
6     public void billing() {
7         System.out.println("BMW -- billing");
8     }
9
10

```

Class can be public, abstract, final only.

Class without any specifier-

Allowed but not a default class.

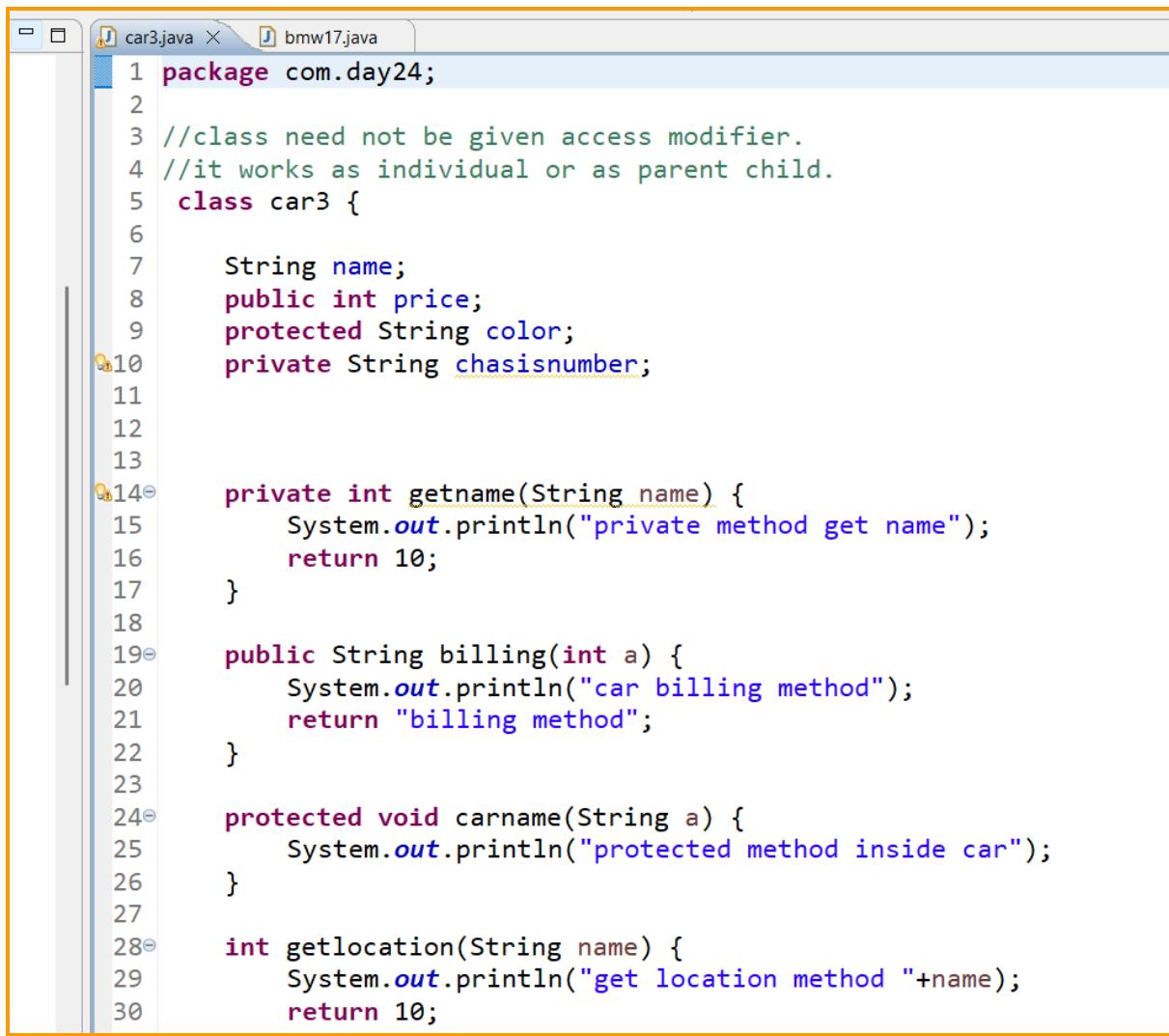


```

actor Navigate Search Project Scout Run Window Help
workspace1 - May2024JavaSession1src/AccMod1/C
Car.java BMW.java Cycle.java Audi.java Truck.java
1 package AccMod1;
2
3 class Car {
4
5     // data members: vars + methods: rule will be same
6     String name;
7     public int price;
8     protected String color;
9     private String chasisNum;
10

```

paste bmw17, car3-



```

car3.java x bmw17.java
1 package com.day24;
2
3 //class need not be given access modifier.
4 //it works as individual or as parent child.
5 class car3 {
6
7     String name;
8     public int price;
9     protected String color;
10    private String chasisnumber;
11
12
13
14    private int getname(String name) {
15        System.out.println("private method get name");
16        return 10;
17    }
18
19    public String billing(int a) {
20        System.out.println("car billing method");
21        return "billing method";
22    }
23
24    protected void carname(String a) {
25        System.out.println("protected method inside car");
26    }
27
28    int getlocation(String name) {
29        System.out.println("get location method "+name);
30        return 10;

```

```

29         System.out.println("get location method "+name);
30     return 10;
31 }
32
33 }
34

```

bmw17.java

```

1 package com.day24;
2
3 //class need not be given any access modifier its ok.
4 class bmw17 extends car3{
5
6     public static void main(String[] args) {
7
8
9         bmw17 b1=new bmw17();
10        int s1=b1.price;
11        System.out.println(s1);
12        String s2=b1.billing(10);
13        System.out.println(s2);
14    }
15
16
17    //public can override the public from parent class.
18    @Override
19    public String billing(int a) {
20        System.out.println("child class bmw billing method " +a);
21        return "billing method";
22    }
23
24 }
25

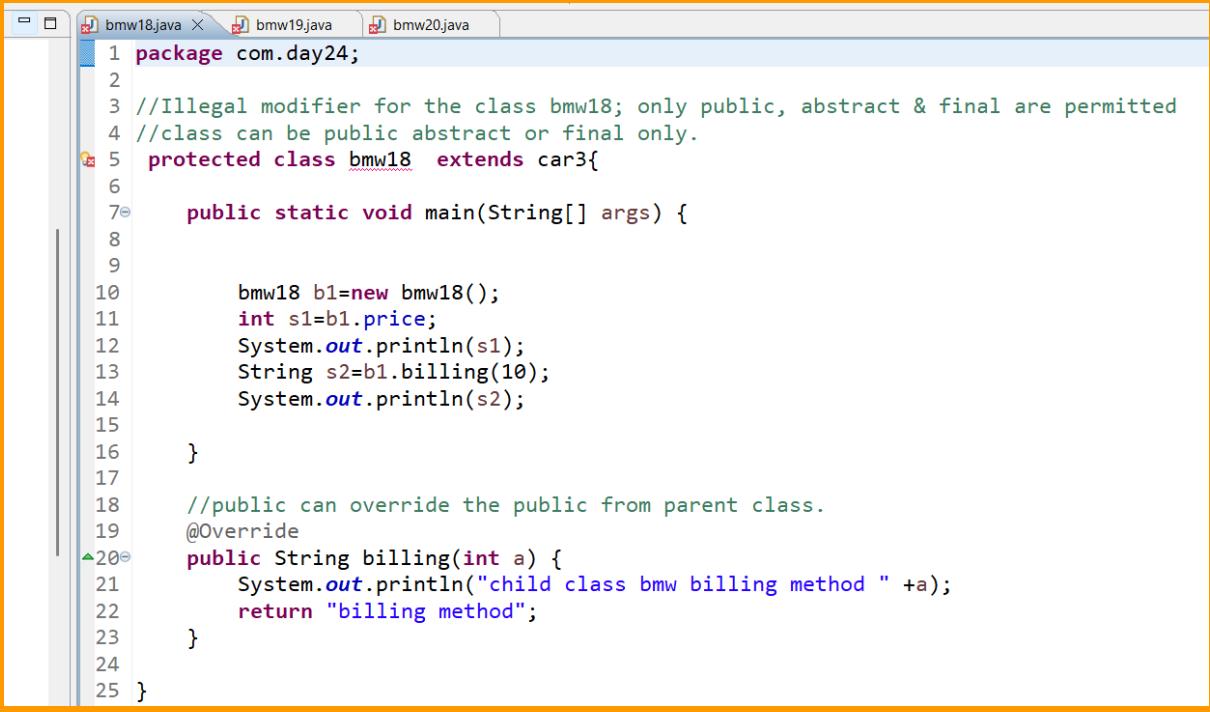
```

```

23
24 }
25
26 //0
27 //child class bmw billing method 10
28 //billing method
29

```

paste bmw18-

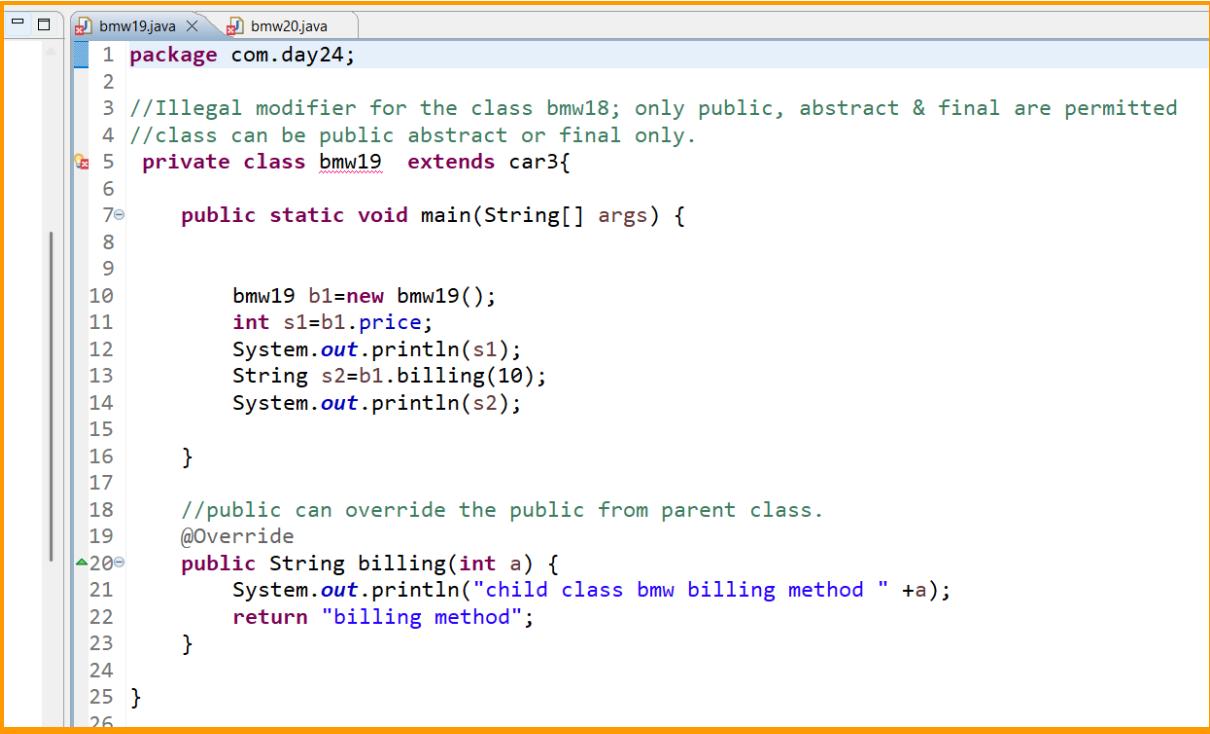


```

1 package com.day24;
2
3 //Illegal modifier for the class bmw18; only public, abstract & final are permitted
4 //class can be public abstract or final only.
5 protected class bmw18 extends car3{
6
7     public static void main(String[] args) {
8
9
10        bmw18 b1=new bmw18();
11        int s1=b1.price;
12        System.out.println(s1);
13        String s2=b1.billing(10);
14        System.out.println(s2);
15
16    }
17
18    //public can override the public from parent class.
19    @Override
20    public String billing(int a) {
21        System.out.println("child class bmw billing method " +a);
22        return "billing method";
23    }
24
25 }

```

paste bmw19-



```

1 package com.day24;
2
3 //Illegal modifier for the class bmw18; only public, abstract & final are permitted
4 //class can be public abstract or final only.
5 private class bmw19 extends car3{
6
7     public static void main(String[] args) {
8
9
10        bmw19 b1=new bmw19();
11        int s1=b1.price;
12        System.out.println(s1);
13        String s2=b1.billing(10);
14        System.out.println(s2);
15
16    }
17
18    //public can override the public from parent class.
19    @Override
20    public String billing(int a) {
21        System.out.println("child class bmw billing method " +a);
22        return "billing method";
23    }
24
25 }
26

```

paste bmw20-

```

1 package com.day24;
2
3 //Syntax error on token "default", delete this token
4 //class can be public abstract or final only.
5 default class bmw20 extends car3{
6
7     public static void main(String[] args) {
8
9         bmw20 b1=new bmw20();
10        int s1=b1.price;
11        System.out.println(s1);
12        String s2=b1.billing(10);
13        System.out.println(s2);
14
15    }
16
17
18    //public can override the public from parent class.
19    @Override
20    public String billing(int a) {
21        System.out.println("child class bmw billing method " +a);
22        return "billing method";
23    }
24
25 }
26
  
```

~~Cannot make child class of the above class~~  
now its allowed.

```

1 package AccMod2;
2
3 import AccMod1.Car;
4
5 public class Audi extends Car{
6
7     public static void main(){
8
9
10        Audi a = new Audi();
11
12
13    }
14
  
```

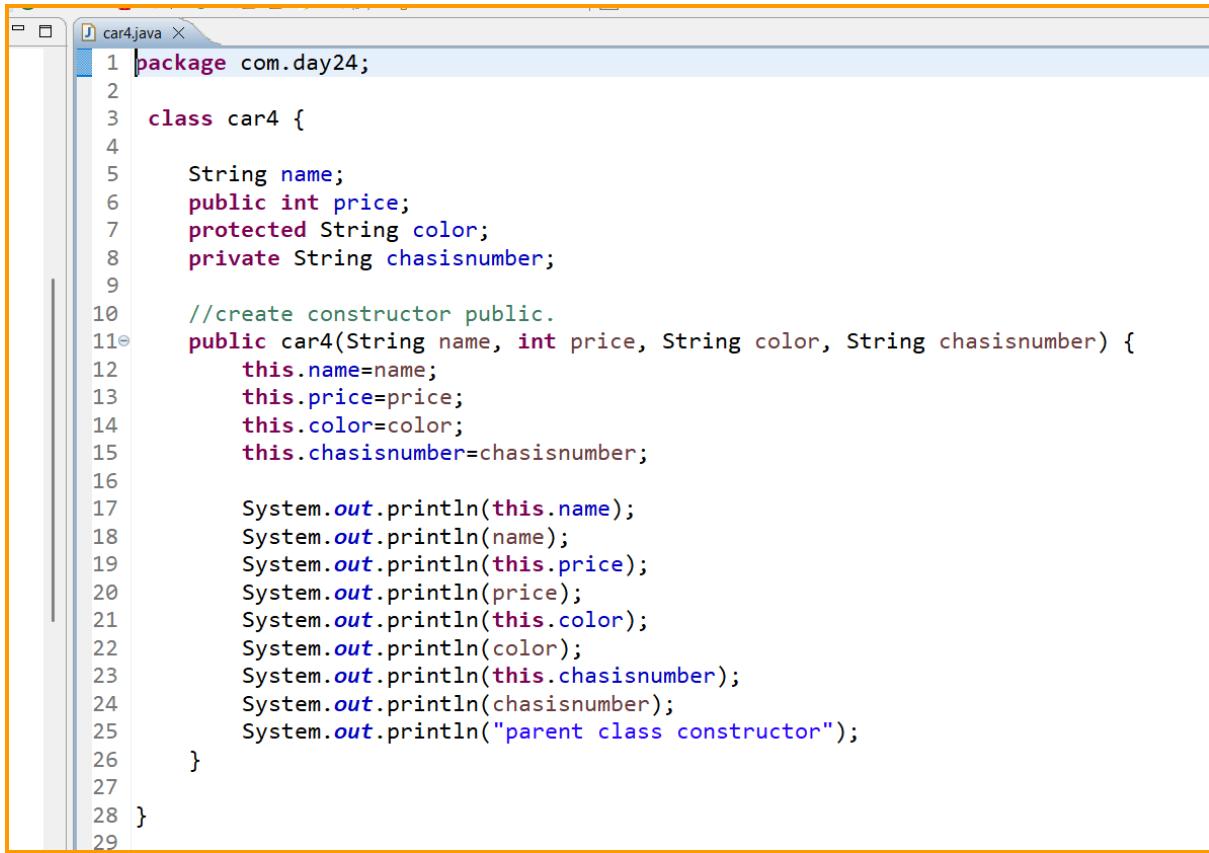
Car cannot be resolved to a type  
8 quick fixes available:

- Change to 'Call' (idk.nashorn.internal.codegen.CompilerConstants)
- Change to 'Card' (ajax.smartcardig)
- Change to 'InheritanceConcept.Car' (InheritanceConcept)
- Change to 'SuperKeyword.Car' (SuperKeyword)
- Create class 'Car'
- Change to 'constructorConcept.Car' (constructorConcept)

Access modifier for constructor, variables and methods only.

## Constructor-

paste car4-

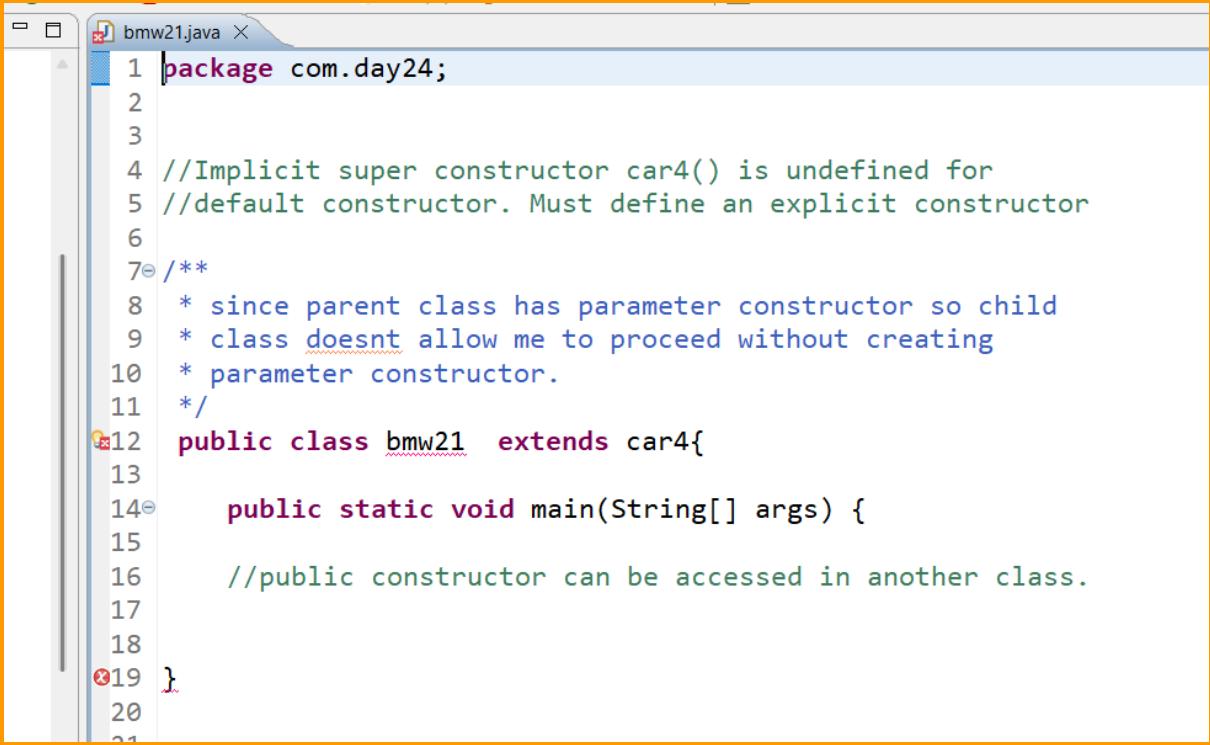


The screenshot shows a Java code editor window titled "car4.java". The code defines a class named "car4" with the following structure:

```
1 package com.day24;
2
3 class car4 {
4
5     String name;
6     public int price;
7     protected String color;
8     private String chasisnumber;
9
10    //create constructor public.
11    public car4(String name, int price, String color, String chasisnumber) {
12        this.name=name;
13        this.price=price;
14        this.color=color;
15        this.chasisnumber=chasisnumber;
16
17        System.out.println(this.name);
18        System.out.println(name);
19        System.out.println(this.price);
20        System.out.println(price);
21        System.out.println(this.color);
22        System.out.println(color);
23        System.out.println(this.chasisnumber);
24        System.out.println(chasisnumber);
25        System.out.println("parent class constructor");
26    }
27
28 }
29
```

The code uses standard Java syntax, including packages, classes, variables, and a constructor. It also includes several print statements to demonstrate the output of the constructor parameters.

paste bmw21-

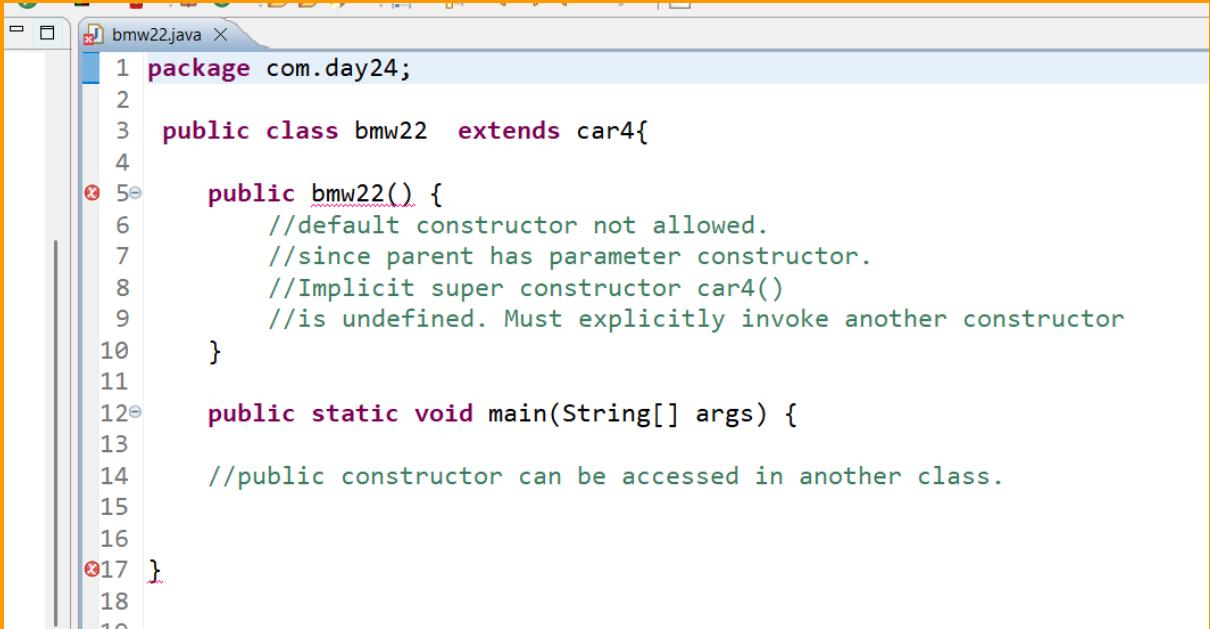


```

1 package com.day24;
2
3
4 //Implicit super constructor car4() is undefined for
5 //default constructor. Must define an explicit constructor
6
7 /**
8  * since parent class has parameter constructor so child
9  * class doesn't allow me to proceed without creating
10 * parameter constructor.
11 */
12 public class bmw21 extends car4{
13
14     public static void main(String[] args) {
15
16         //public constructor can be accessed in another class.
17
18
19    }
20
21

```

paste bmw22-



```

1 package com.day24;
2
3 public class bmw22 extends car4{
4
5     public bmw22() {
6         //default constructor not allowed.
7         //since parent has parameter constructor.
8         //Implicit super constructor car4()
9         //is undefined. Must explicitly invoke another constructor
10    }
11
12    public static void main(String[] args) {
13
14        //public constructor can be accessed in another class.
15
16
17    }
18
19

```

paste bmw23-

```
bmw23.java
1 package com.day24;
2
3
4 public class bmw23 extends car4{
5
6     public bmw23() {
7         super("bmw", 10000, "black", "1234");
8         //this is one way to resolve.
9     }
10
11    public static void main(String[] args) {
12
13        //public constructor can be accessed in another class.
14
15    }
16}
17}
```

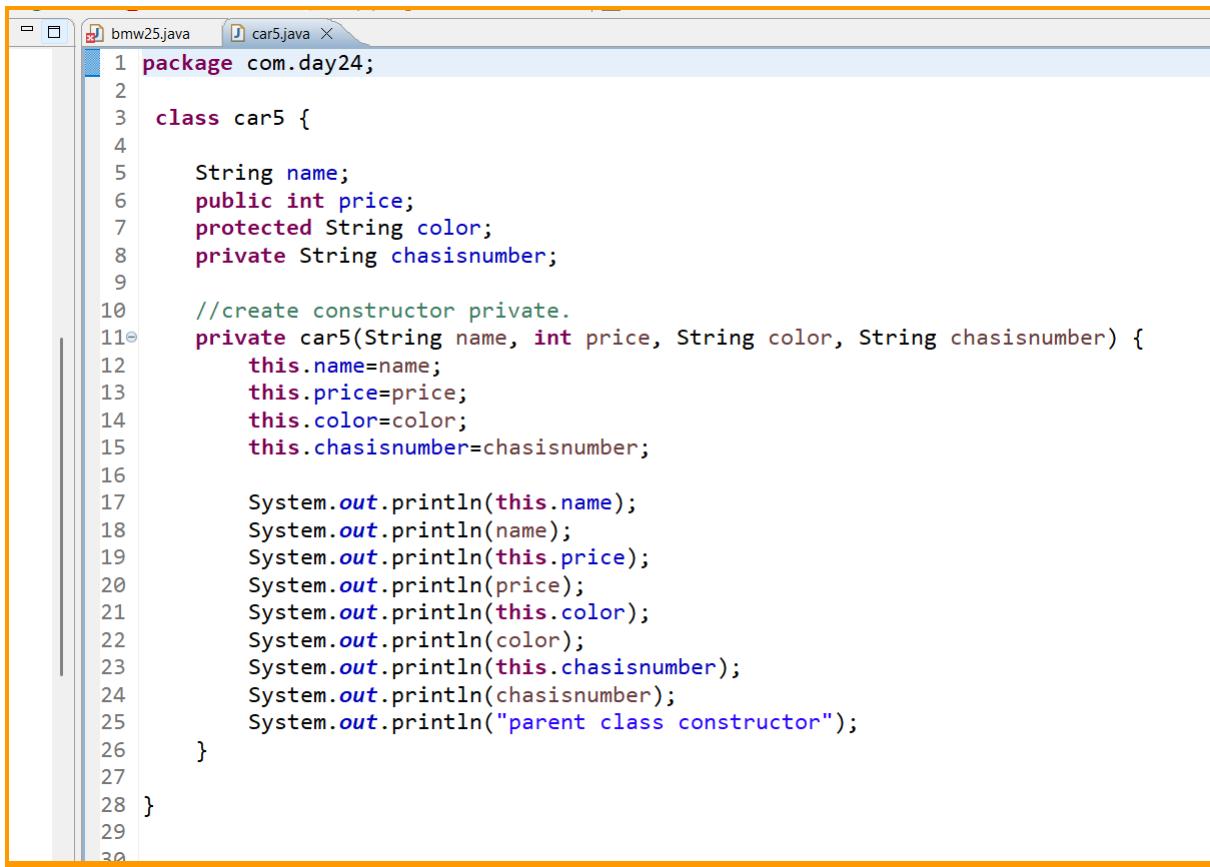
paste bmw24-

The screenshot shows a Java code editor with the file `bmw24.java` open. The code demonstrates inheritance and constructor resolution. It defines a class `bmw24` that extends `car4`. The `bmw24` class has its own constructor that takes `name`, `price`, `color`, `chasisnumber`, and `address` as parameters. It also has a `phone` variable. The constructor calls the parent class's constructor using `super(name, price, color, chasisnumber)`. The code then prints out the values of `name`, `price`, `color`, `chasisnumber`, `address`, and `phone` using `System.out.println`. A comment indicates that the field `chasisnumber` is not visible from the child class due to scoping rules.

```
bmw24.java
1 package com.day24;
2
3
4 public class bmw24 extends car4{
5
6     //this is another way to resolve.
7     //create the same constructor in child and add more variables if needed.
8     String address;
9     int phone;
10
11    public bmw24(String name, int price, String color, String chasisnumber, String address
12                  ,int phone) {
13        super(name, price, color, chasisnumber);
14        System.out.println(this.name);
15        System.out.println(name);
16        System.out.println(this.price);
17        System.out.println(price);
18        System.out.println(this.color);
19        System.out.println(color);
20    //    System.out.println(this.chasisnumber); //The field car4.chasisnumber is not visible
21        System.out.println(chasisnumber);
22        System.out.println("child class constructor");
23        System.out.println(this.address);
24        System.out.println(address);
25        System.out.println(this.phone);
26        System.out.println(phone);
27    }
}
```

```
26         System.out.println(phone);
27     }
28
29 public static void main(String[] args) {
30
31     //public constructor can be accessed in another class.
32     car4 c1=new car4("karan", 23434, "green", "@#4324");
33     System.out.println(c1);
34
35 }
36 }
37
38
39
40 // karan
41 // karan
42 // 23434
43 // 23434
44 // green
45 // green
46 // #@4324
47 // #@4324
48 // parent class constructor
49 // com.day24.car4@24d46ca6
50
```

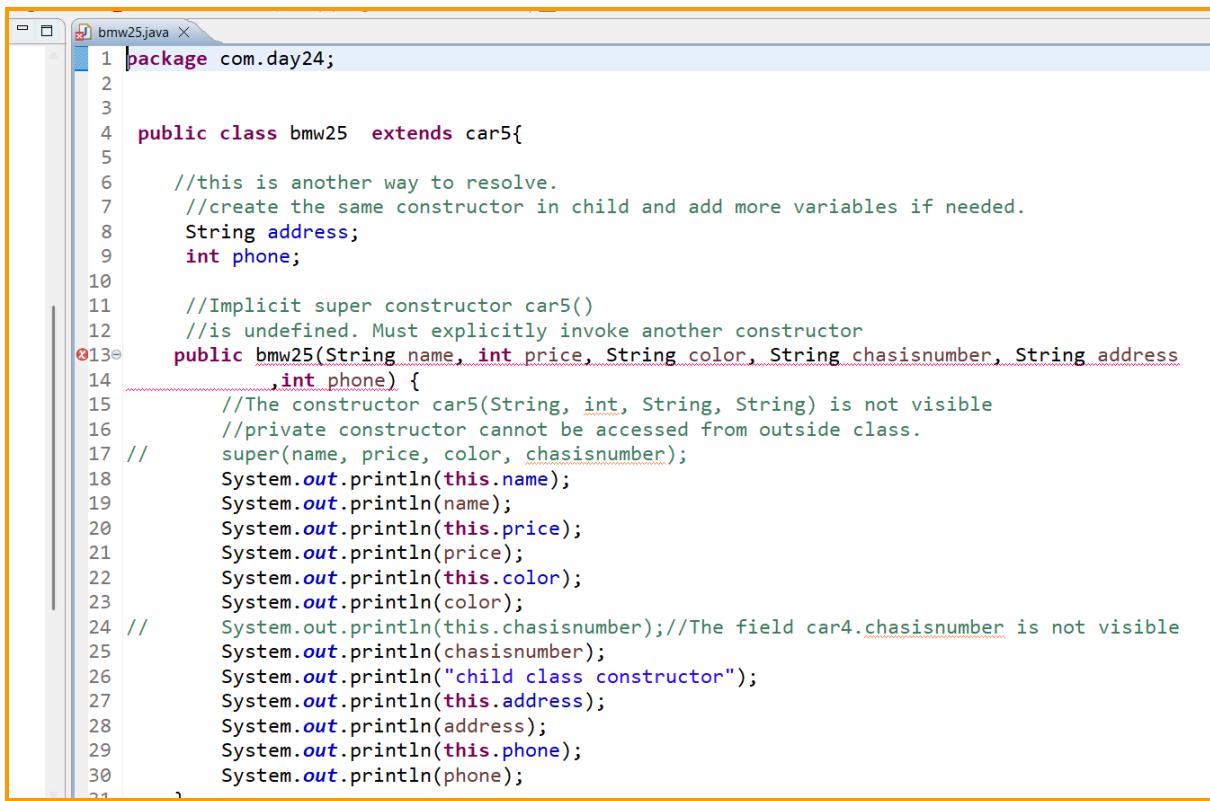
paste car5. bmw25-



```

1 package com.day24;
2
3 class car5 {
4
5     String name;
6     public int price;
7     protected String color;
8     private String chasisnumber;
9
10    //create constructor private.
11    private car5(String name, int price, String color, String chasisnumber) {
12        this.name=name;
13        this.price=price;
14        this.color=color;
15        this.chasisnumber=chasisnumber;
16
17        System.out.println(this.name);
18        System.out.println(name);
19        System.out.println(this.price);
20        System.out.println(price);
21        System.out.println(this.color);
22        System.out.println(color);
23        System.out.println(this.chasisnumber);
24        System.out.println(chasisnumber);
25        System.out.println("parent class constructor");
26    }
27
28 }
29
30

```



```

1 package com.day24;
2
3
4 public class bmw25  extends car5{
5
6     //this is another way to resolve.
7     //create the same constructor in child and add more variables if needed.
8     String address;
9     int phone;
10
11     //Implicit super constructor car5()
12     //is undefined. Must explicitly invoke another constructor
13     public bmw25(String name, int price, String color, String chasisnumber, String address,
14                  int phone) {
15         //The constructor car5(String, int, String, String) is not visible
16         //private constructor cannot be accessed from outside class.
17         //super(name, price, color, chasisnumber);
18         System.out.println(this.name);
19         System.out.println(name);
20         System.out.println(this.price);
21         System.out.println(price);
22         System.out.println(this.color);
23         System.out.println(color);
24         System.out.println(this.chasisnumber);//The field car4.chasisnumber is not visible
25         System.out.println(chasisnumber);
26         System.out.println("child class constructor");
27         System.out.println(this.address);
28         System.out.println(address);
29         System.out.println(this.phone);
30         System.out.println(phone);
31

```

```

29         System.out.println(this.phone);
30         System.out.println(phone);
31     }
32
33@ public static void main(String[] args) {
34
35     //private constructor cannot be accessed in another class.
36     //The constructor car5(String, int, String, String) is not visible
37     car5 c1=new car5("karan", 23434, "green", "@#4324");
38     System.out.println(c1);
39
40 }
41
42

```

paste car6, bmw26-

```

bmw26.java car6.java
1 package com.day24;
2
3 class car6 {
4
5     String name;
6     public int price;
7     protected String color;
8     private String chasisnumber;
9
10    //create constructor protected.
11@ protected car6(String name, int price, String color, String chasisnumber) {
12     this.name=name;
13     this.price=price;
14     this.color=color;
15     this.chasisnumber=chasisnumber;
16
17     System.out.println(this.name);
18     System.out.println(name);
19     System.out.println(this.price);
20     System.out.println(price);
21     System.out.println(this.color);
22     System.out.println(color);
23     System.out.println(this.chasisnumber);
24     System.out.println(chasisnumber);
25     System.out.println("parent class constructor");
26 }
27
28 }
29
30

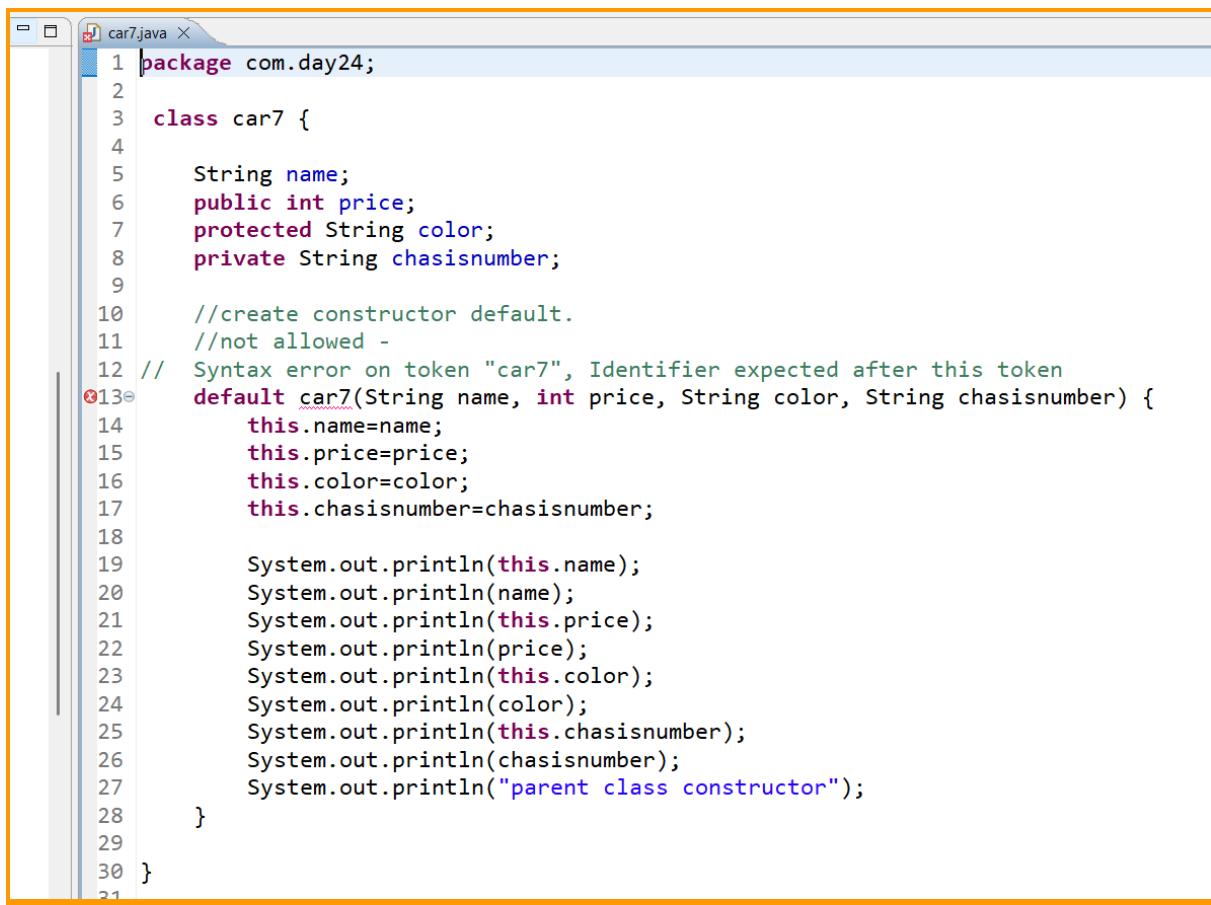
```

```

1 package com.day24;
2
3
4 public class bmw26 extends car6{
5
6     //this is another way to resolve.
7     //create the same constructor in child and add more variables if needed.
8     String address;
9     int phone;
10
11    //protected constructor can be accessed in child class.
12    public bmw26(String name, int price, String color, String chasisnumber, String address
13                 ,int phone) {
14        super(name, price, color, chasisnumber);
15        System.out.println(this.name);
16        System.out.println(name);
17        System.out.println(this.price);
18        System.out.println(price);
19        System.out.println(this.color);
20        System.out.println(color);
21        // System.out.println(this.chasisnumber); //The field car4.chasisnumber is not visible
22        System.out.println(chasisnumber);
23        System.out.println("child class constructor");
24        System.out.println(this.address);
25        System.out.println(address);
26        System.out.println(this.phone);
27        System.out.println(phone);
28    }
29
30    public static void main(String[] args) {
31
32        //protected constructor can be accessed in another class.
33        car6 c1=new car6("karan", 23434, "green", "@#4324");
34        System.out.println(c1);
35
36    }
37}
38
39 // karan
40 // karan
41 // 23434
42 // 23434
43 // green
44 // green
45 // #@#4324
46 // #@#4324
47 // parent class constructor
48 // com.day24.car4@4517d9a3
49
50

```

paste car7-



```

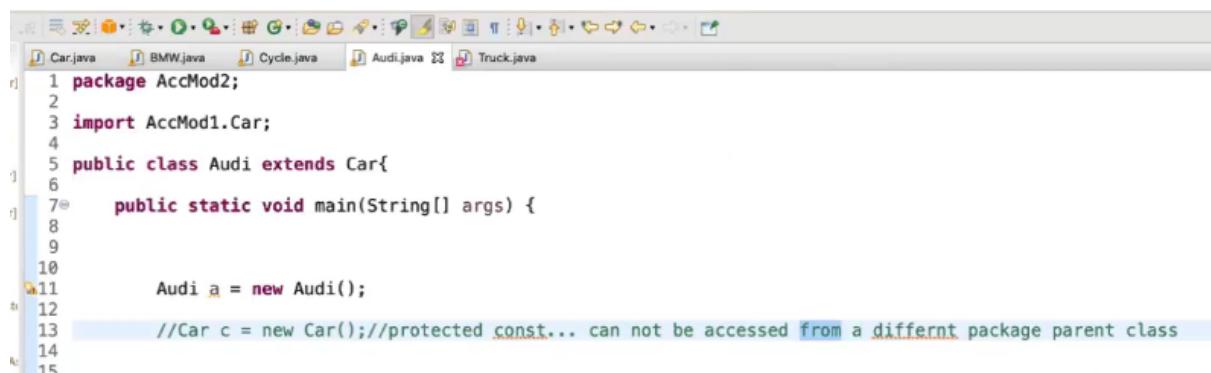
1 package com.day24;
2
3 class car7 {
4
5     String name;
6     public int price;
7     protected String color;
8     private String chasisnumber;
9
10    //create constructor default.
11    //not allowed -
12 // Syntax error on token "car7", Identifier expected after this token
13@   default car7(String name, int price, String color, String chasisnumber) {
14     this.name=name;
15     this.price=price;
16     this.color=color;
17     this.chasisnumber=chasisnumber;
18
19     System.out.println(this.name);
20     System.out.println(name);
21     System.out.println(this.price);
22     System.out.println(price);
23     System.out.println(this.color);
24     System.out.println(color);
25     System.out.println(this.chasisnumber);
26     System.out.println(chasisnumber);
27     System.out.println("parent class constructor");
28   }
29
30 }
31

```

Can be public, private, protected, default.

Protected constructor cannot be accessed in different package child class also-

//car6 cannot be resolved to a type



```

1 package AccMod2;
2
3 import AccMod1.Car;
4
5 public class Audi extends Car{
6
7@   public static void main(String[] args) {
8
9
10
11     Audi a = new Audi();
12
13     //Car c = new Car(); //protected const... can not be accessed from a differnt package parent class
14
15

```