



TYPE CASTING IN JAVA



BY

MUNI AMALA TELU

TYPE CASTING IN JAVA

Type casting:

Convert a value from one data type to another data type is known as type casting.

Two types of type casting:

1. Implicit type casting:

Converting a lower data type into a higher one is called implicit type casting. It is also known as widening. It is done automatically. It is safe because there is no chance to lose data.

2. Explicit type casting:

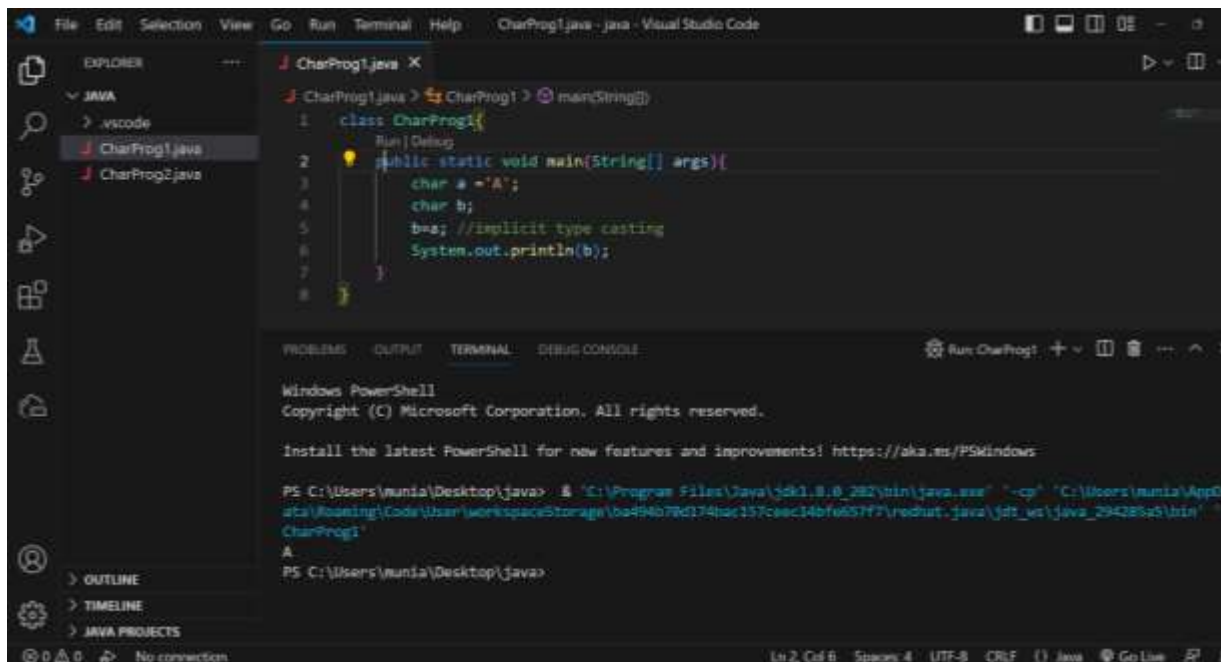
Converting a higher data type into a lower one is called explicit type casting. It is also known as narrowing. It is done manually by the programmer. If we do not perform casting then the compiler reports a compile-time error.

Program 1: (char to char)

```
class CharProg1 {  
    public static void main(String[] args){  
        char a ='A';  
        char b;  
        b=a; //implicit type casting  
        System.out.println(b);  
    }  
}
```

Casting Not Required

Output:



```
File Edit Selection View Go Run Terminal Help CharProg1.java - java - Visual Studio Code
EXPLORER
  JAVA
    .vscode
    CharProg1.java
    CharProg2.java
  CharProg1.java X
CharProg1.java > CharProg1 > main(String[])
1 class CharProg1{
2   public static void main(String[] args){
3     char a = 'A';
4     char b;
5     b=a; //implicit type casting
6     System.out.println(b);
7   }
8 }
Run | Debug
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

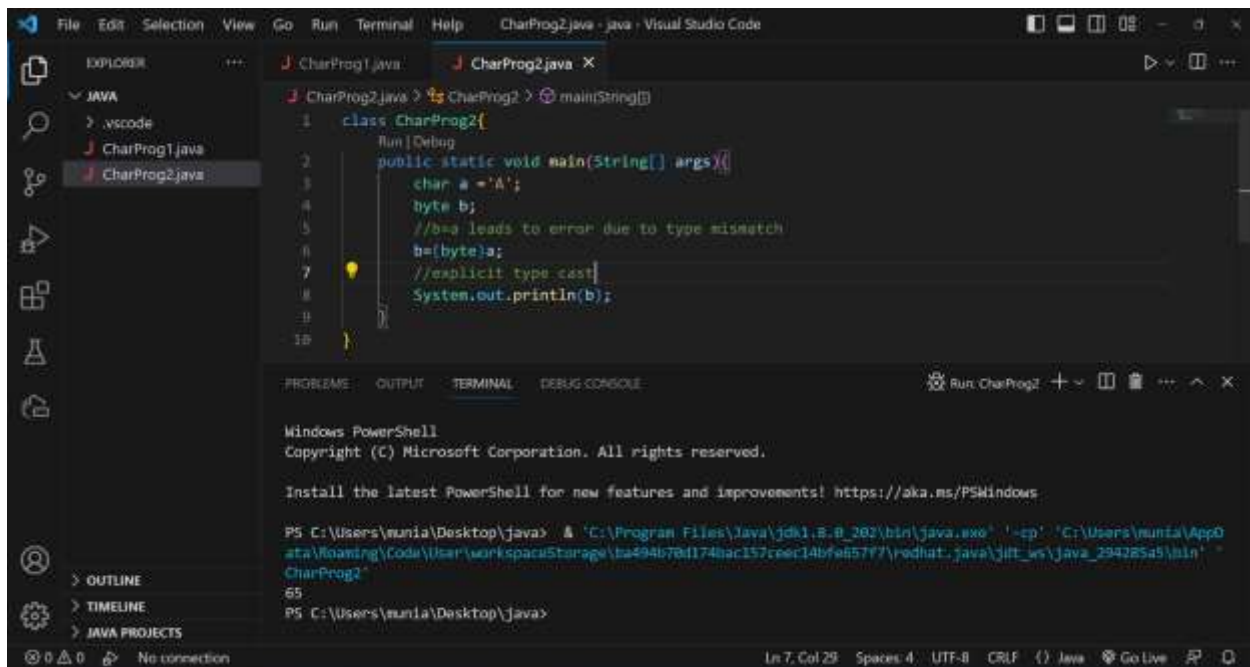
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Local\Microsoft\Windows\Code\workspaceStorage\ba494b78d174b8c157c8ec14bfe657f7\redhut.java\jdk1_8\java_294285x5\bin" "CharProg1"
A
PS C:\Users\munia\Desktop\java>
```

Program 2: (char to byte)

```
class CharProg2{
    public static void main(String[] args){
        char a ='A';
        byte b;
        //b=a leads to error due to type mismatch
        b=(byte)a;
        //explicit type cast
        System.out.println(b);
    }
}
```

Explicit type casting

Output:

A screenshot of the Visual Studio Code editor. The Explorer pane on the left shows a project with two Java files: CharProg1.java and CharProg2.java. The main editor window displays CharProg2.java with the following code:

```
1 class CharProg2{
2     public static void main(String[] args){
3         char a = 'A';
4         byte b;
5         //b=a leads to error due to type mismatch
6         b=(byte)a;
7         //explicit type cast
8         System.out.println(b);
9     }
10 }
```

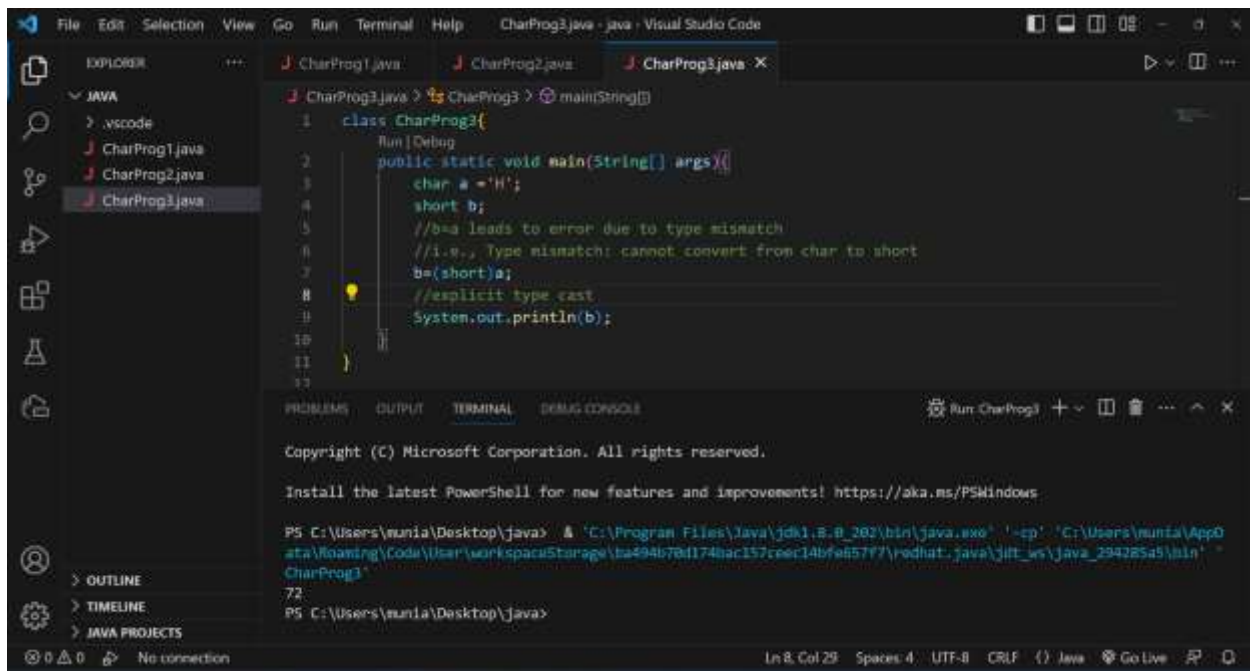
The bottom panel shows the TERMINAL with a Windows PowerShell prompt. The command executed is: `PS C:\Users\munia\Desktop\java> java -cp 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Local\Roaming\Code\User\workspaceStorage\ba494b7bd174bac157ceec14bfe657f7\redhat.java\jdt_ws\java_294285a5\bin' CharProg2`. The output is `65`. The status bar at the bottom indicates 'Ln 7, Col 29', 'Spaces: 4', 'UTF-8', 'CRLF', and 'Java'.

Program 3: (char to short)

```
class CharProg3{
    public static void main(String[] args){
        char a ='H';
        short b;
        //b=a leads to error due to type mismatch
        //i.e., Type mismatch: cannot convert from char to short
        b=(short)a;
        //explicit type cast
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named CharProg3.java. The code defines a class CharProg3 with a main method. Inside the main method, a char variable 'a' is assigned the value 'H', and a short variable 'b' is declared. A comment indicates that 'b=a' would lead to a type mismatch error because a char cannot be converted to a short. To resolve this, an explicit type cast is used: b=(short)a;. The program then prints the value of b using System.out.println(b);. The terminal at the bottom shows the command to run the program, which executes successfully.

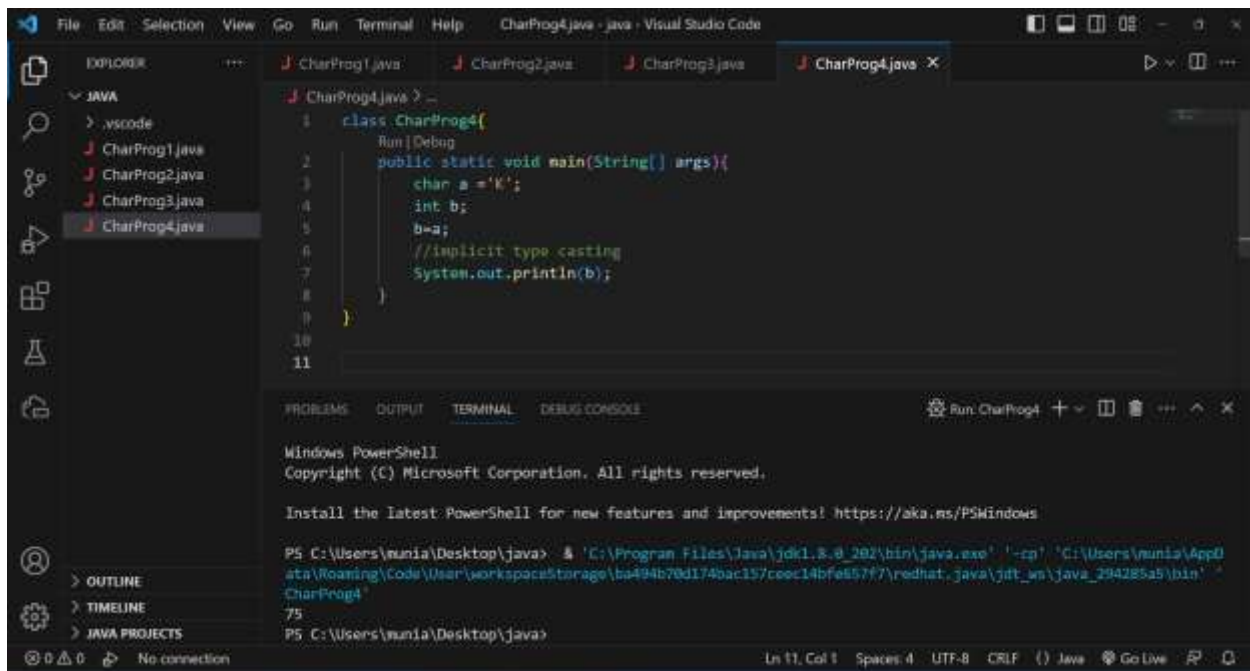
```
1 class CharProg3{
2     public static void main(String[] args){
3         char a = 'H';
4         short b;
5         //b=a leads to error due to type mismatch
6         //i.e., Type mismatch: cannot convert from char to short
7         b=(short)a;
8         //explicit type cast
9         System.out.println(b);
10    }
11 }
```

Program 4: (char to int)

```
class CharProg4{
    public static void main(String[] args){
        char a ='K';
        int b;
        b=a;
        //implicit type casting
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code interface with a Java file named CharProg4.java. The code defines a class CharProg4 with a main method. Inside the main method, a char variable 'a' is assigned the value 'K', an int variable 'b' is declared, 'b' is assigned the value of 'a', and 'b' is printed using System.out.println(b). A comment indicates that this is an example of implicit type casting. The terminal at the bottom shows the command to run the program, which successfully executes and prints the character 'K'.

```
1 class CharProg4{
2     public static void main(String[] args){
3         char a = 'K';
4         int b;
5         b=a;
6         //implicit type casting
7         System.out.println(b);
8     }
9 }
10
11
```

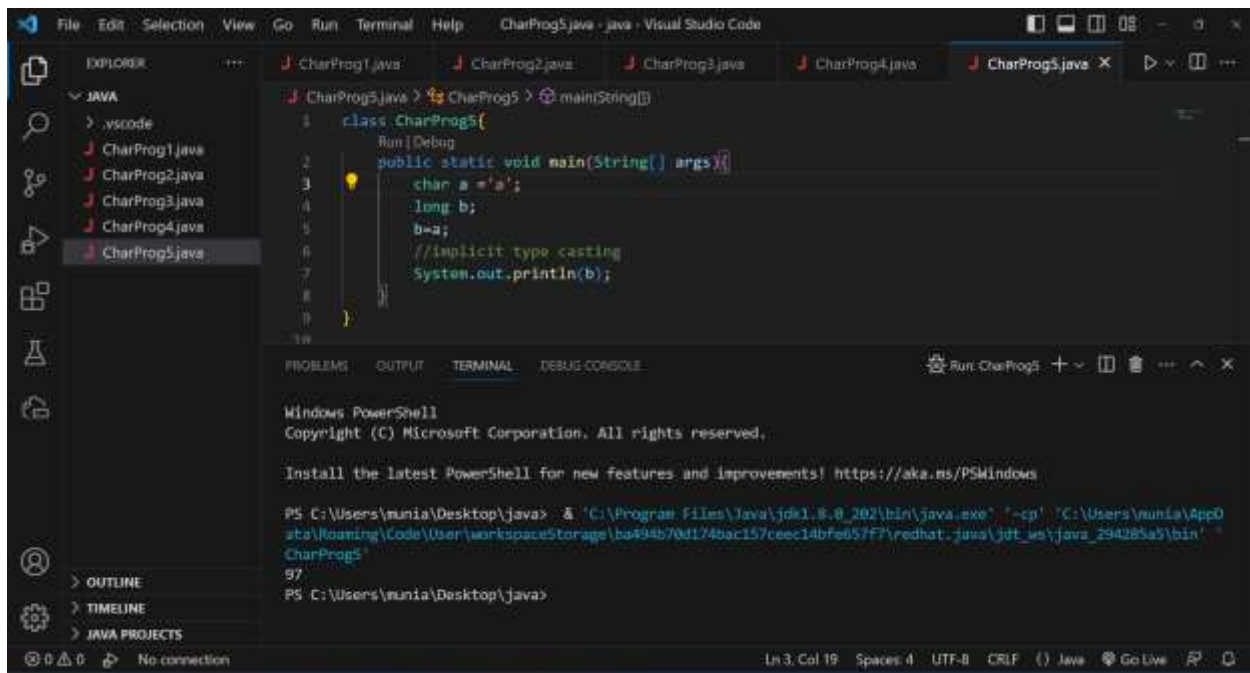
```
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec14bfa657f7\redhat_java\jdt_ws\java_294285a5\bin' 'CharProg4'
75
PS C:\Users\munia\Desktop\java>
```

Program 5: (char to long)

```
class CharProg5{
    public static void main(String[] args){
        char a ='a';
        long b;
        b=a;
        //implicit type casting
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



```
class CharProg5{
    public static void main(String[] args){
        char a = 'a';
        long b;
        b=a;
        //implicit type casting
        System.out.println(b);
    }
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

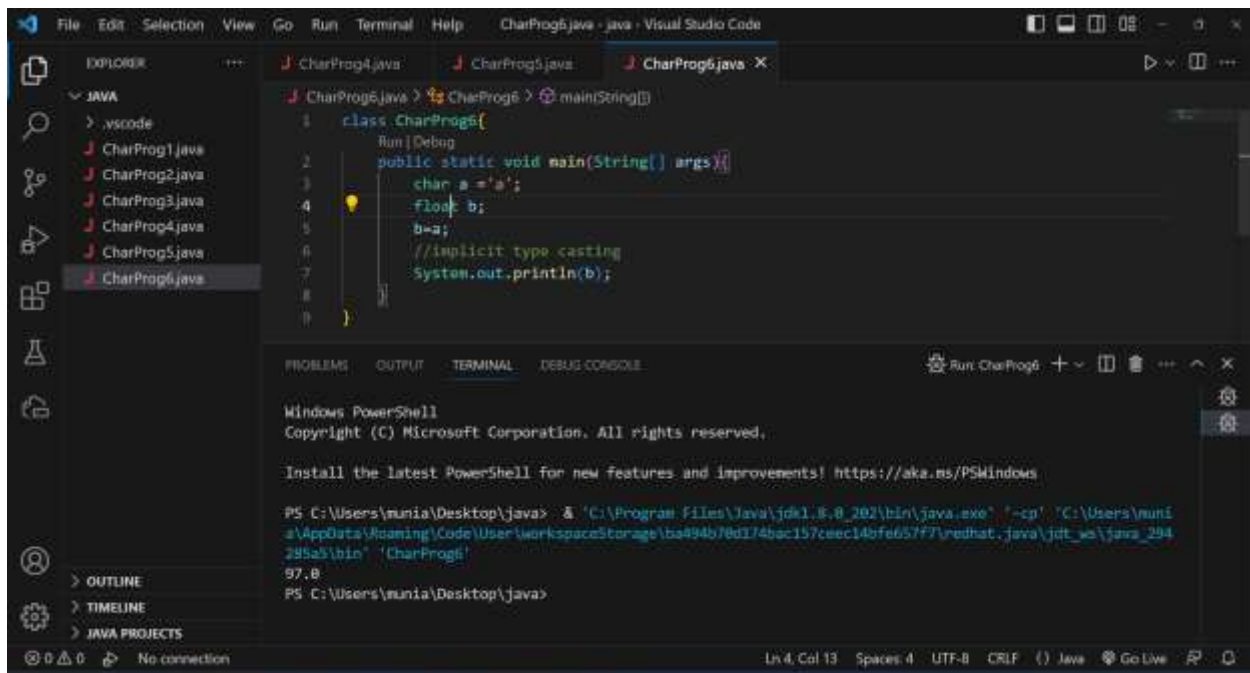
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" "-cp" "C:\Users\munia\AppData\Local\Roaming\Code\User\workspaceStorage\b494b7bd174bac157ceec14bfe657f7\redhat.java\jdt_ws\java_294285a5\bin" "CharProg5"
97
PS C:\Users\munia\Desktop\java>

Program 6: (char to float)

```
class CharProg6{
    public static void main(String[] args){
        char a ='a';
        float b;
        b=a;
        //implicit type casting
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named CharProg6.java. The code defines a class CharProg6 with a main method. Inside the main method, a char variable 'a' is assigned the value 'a', a float variable 'b' is declared, 'b' is assigned the value of 'a', and 'b' is printed using System.out.println(b). A comment indicates this is implicit type casting. The terminal at the bottom shows the command to run the program and the output '97.8'.

```
class CharProg6 {  
    public static void main(String[] args) {  
        char a = 'a';  
        float b;  
        b = a;  
        //implicit type casting  
        System.out.println(b);  
    }  
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

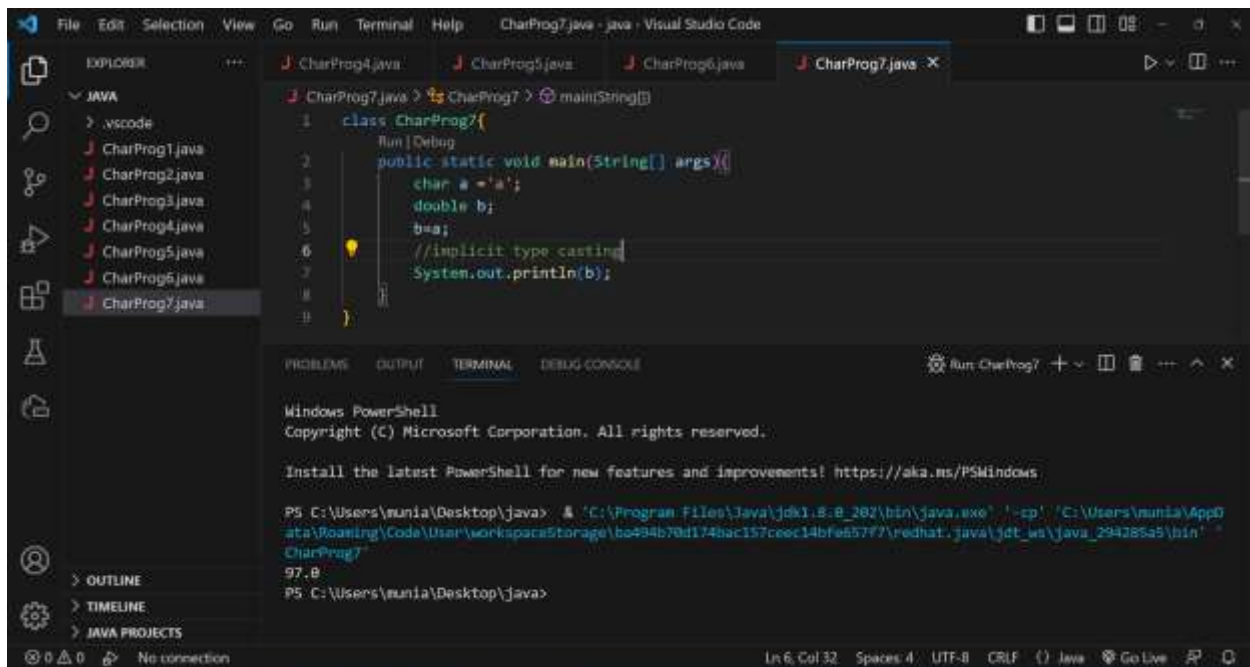
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba894b70d174bac157ceec14bfe657f7\redhat.java\jdt_ws\java_294285a3\bin" "CharProg6"
97.8
PS C:\Users\munia\Desktop\java>

Program 7: (char to double)

```
class CharProg7 {  
    public static void main(String[] args) {  
        char a = 'a';  
        double b;  
        b = a;  
        //implicit type casting  
        System.out.println(b);  
    }  
}
```


Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named CharProg7.java. The code defines a class CharProg7 with a main method. Inside the main method, a char variable 'a' is assigned the value 'a', and a double variable 'b' is assigned the value of 'a'. A comment indicates that this is an example of implicit type casting. The output of the program is shown in the terminal window, which displays the value 97.0.

```
class CharProg7 {  
    public static void main(String[] args) {  
        char a = 'a';  
        double b;  
        b = a;  
        //implicit type casting  
        System.out.println(b);  
    }  
}
```

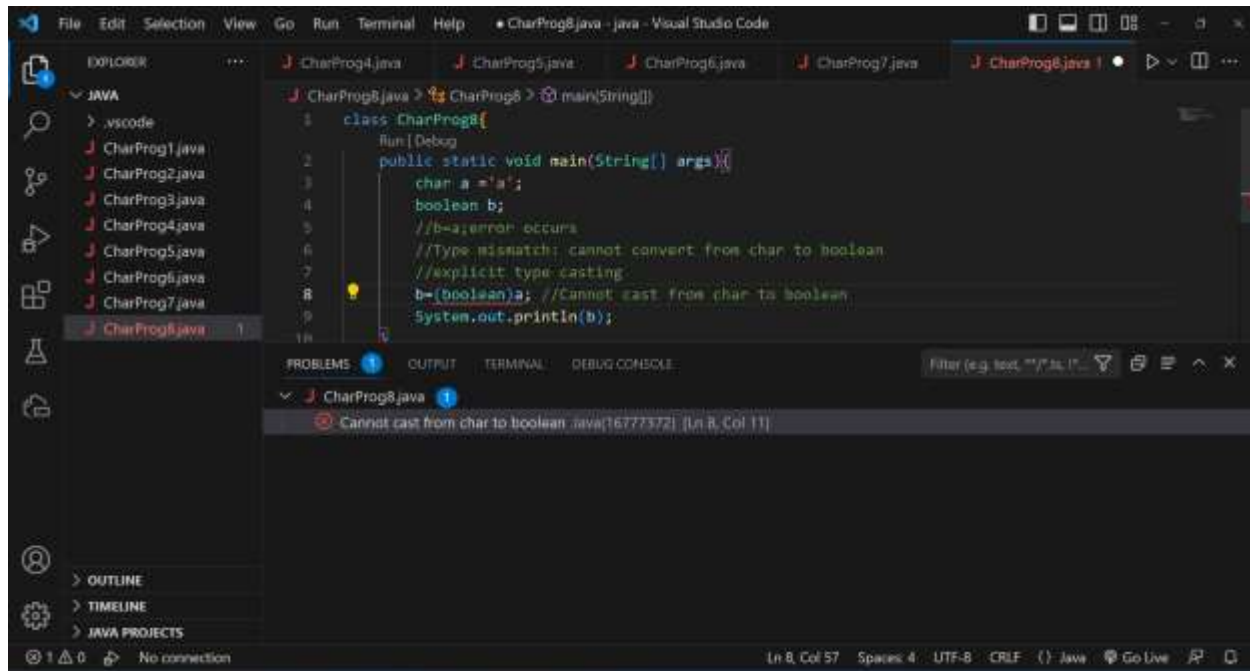
```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Local\Roaming\Code\User\workspaceStorage\ba94b79d174bac157ceec14bfe657f7\redhat.java\jdk_ws\java_294285a5\bin" "CharProg7"  
97.0  
PS C:\Users\munia\Desktop\java>
```

Program 8: (char to boolean)

```
class CharProg8 {  
    public static void main(String[] args) {  
        char a = 'a';  
        boolean b;  
        //b=a;error occurs  
        //Type mismatch: cannot convert from char to boolean  
        //explicit type casting  
        b=(boolean)a; //Cannot cast from char to boolean  
        System.out.println(b);  
    }  
}
```

Casting cannot possible

Output:

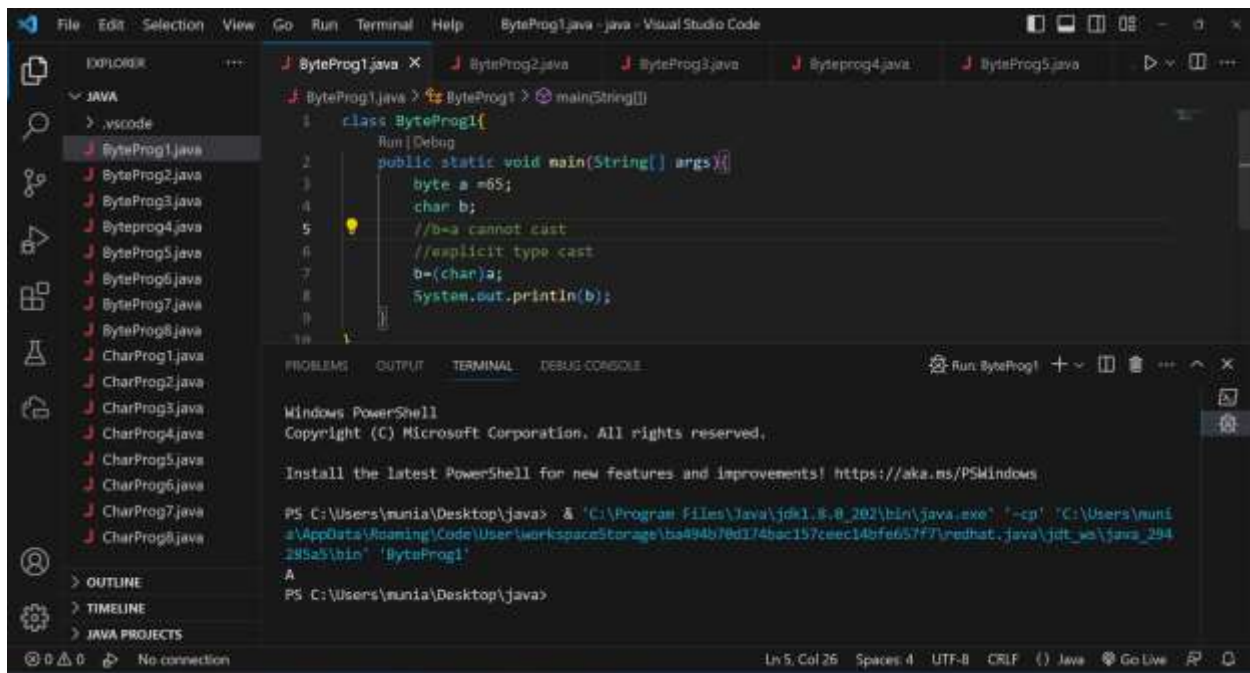


Program 9: (byte to char)

```
class ByteProg1 {  
    public static void main(String[] args) {  
        byte a = 65;  
        char b;  
        //b=a cannot cast  
        //explicit type cast  
        b=(char)a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including ByteProg1.java through ByteProg8.java and CharProg1.java through CharProg8.java. The main editor displays ByteProg1.java with the following code:

```
1 class ByteProg1 {  
2     public static void main(String[] args) {  
3         byte a = 65;  
4         char b;  
5         //b=a cannot cast  
6         //explicit type cast  
7         b=(char)a;  
8         System.out.println(b);  
9     }  
10 }
```

The output pane at the bottom shows the command prompt results:

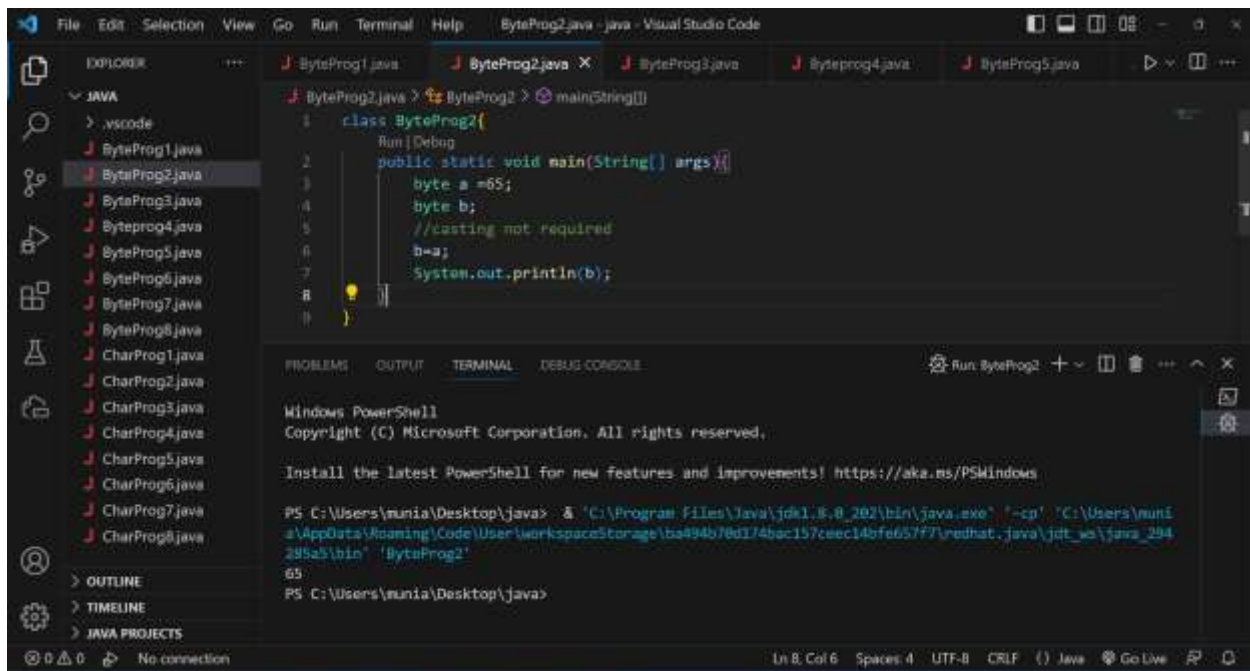
```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' "-cp" 'C:\Users\munia\AppData\Local\Temp\Code\User\workspaceStorage\ba494b7ed174bac157ceec14bfe657f7\redhat.java\jdk_w\java_294285a3\bin' 'ByteProg1'  
A  
PS C:\Users\munia\Desktop\java>
```

Program 10: (byte to byte)

```
class ByteProg2 {  
    public static void main(String[] args) {  
        byte a = 65;  
        byte b;  
        //casting not required  
        b=a;  
        System.out.println(b);  
    }  
}
```

Casting not required

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including ByteProg1.java through ByteProg8.java and CharProg1.java through CharProg8.java. The main editor displays the code for ByteProg2.java, which contains the following code:

```
1 class ByteProg2 {  
2     public static void main(String[] args) {  
3         byte a = 65;  
4         byte b;  
5         //casting not required  
6         b=a;  
7         System.out.println(b);  
8     }  
9 }
```

The Run and Debug toolbar is visible above the code. Below the code editor, the Output pane shows the execution results in a Windows PowerShell window:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b7ed174bac157ceec14bfe657f7\redhat.java\jdt_ws\java_294285a3\bin" "ByteProg2"  
65  
PS C:\Users\munia\Desktop\java>
```

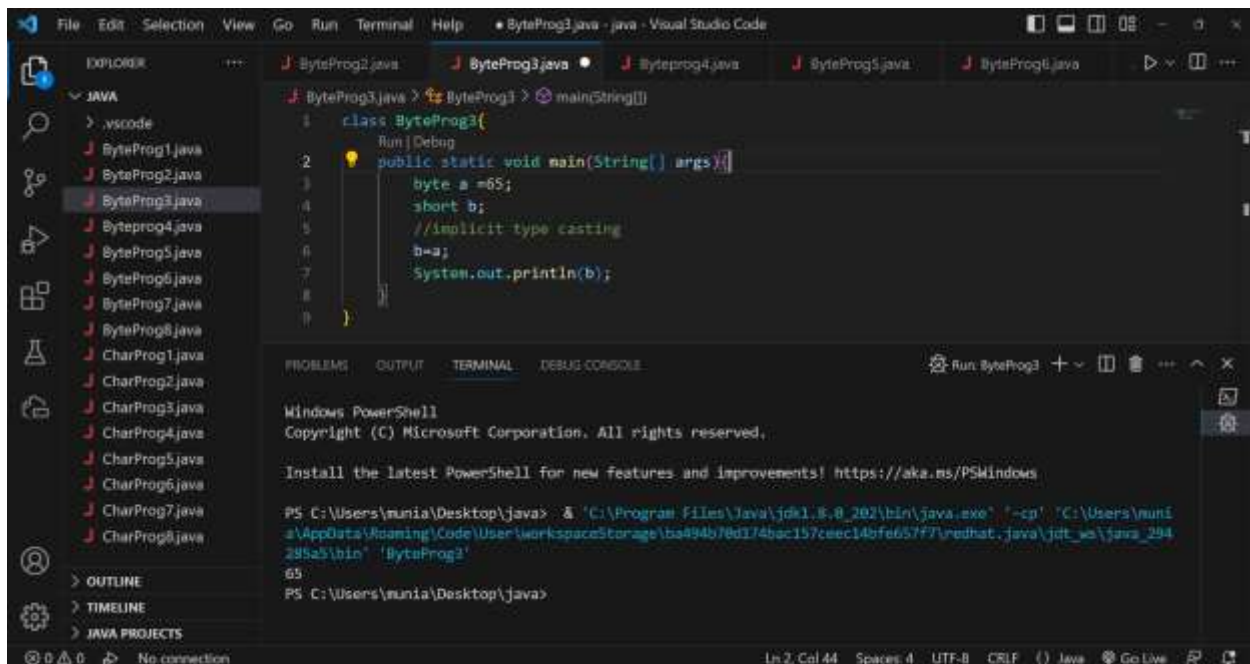
The status bar at the bottom indicates the current line and column (Ln 8, Col 6), the number of spaces (Spaces: 4), the encoding (UTF-8), the line endings (CRLF), the language (Java), and the Go Live button.

Program 11: (byte to short)

```
class ByteProg3 {  
    public static void main(String[] args) {  
        byte a = 65;  
        short b;  
        //implicit type casting  
        b = a;  
        System.out.println(b);  
    }  
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, with ByteProg3.java selected. The Editor pane displays the code for ByteProg3.java, which includes a class definition and a main method. The code is as follows:

```
class ByteProg3{
    public static void main(String[] args){
        byte a = 65;
        short b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

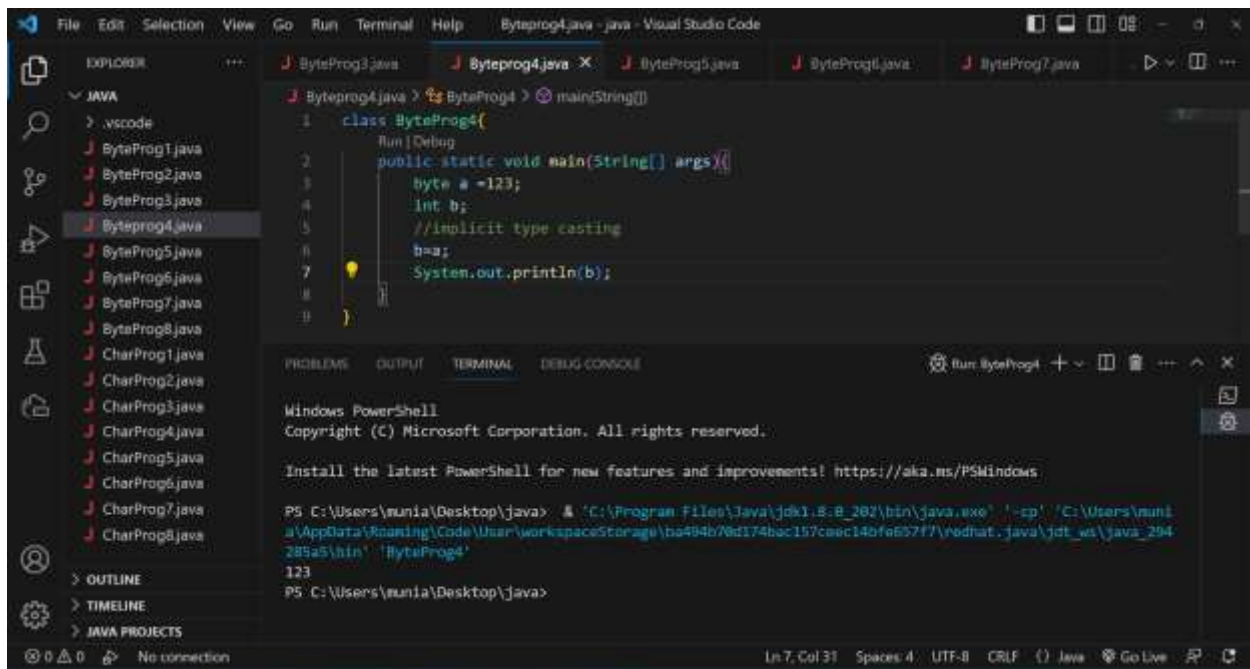
The Output pane at the bottom shows the execution of the program, displaying the number 65.

Program 12: (byte to int)

```
class ByteProg4{
    public static void main(String[] args){
        byte a = 123;
        int b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, with 'Byteprog4.java' selected. The main editor displays the code for 'Byteprog4.java':

```
1 class Byteprog4{
2     public static void main(String[] args){
3         byte a =123;
4         int b;
5         //implicit type casting
6         b=a;
7         System.out.println(b);
8     }
9 }
```

The bottom pane shows the 'TERMINAL' output, which is a Windows PowerShell window. It displays the command to run the program and the output '123':

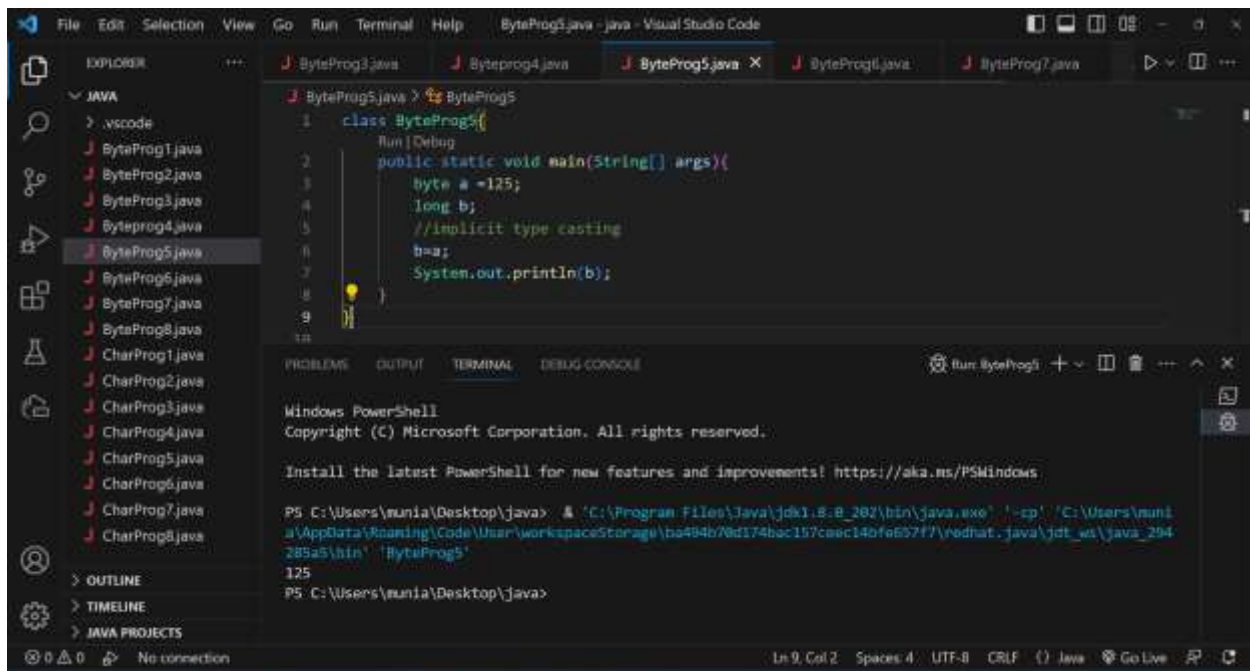
```
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin' 'Byteprog4'
123
PS C:\Users\munia\Desktop\java>
```

Program 13: (byte to long)

```
class ByteProg5{
    public static void main(String[] args){
        byte a =125;
        long b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, with ByteProg5.java selected. The Editor pane displays the following code:

```
1 class ByteProg5 {  
2     public static void main(String[] args) {  
3         byte a = 125;  
4         long b;  
5         //implicit type casting  
6         b = a;  
7         System.out.println(b);  
8     }  
9 }
```

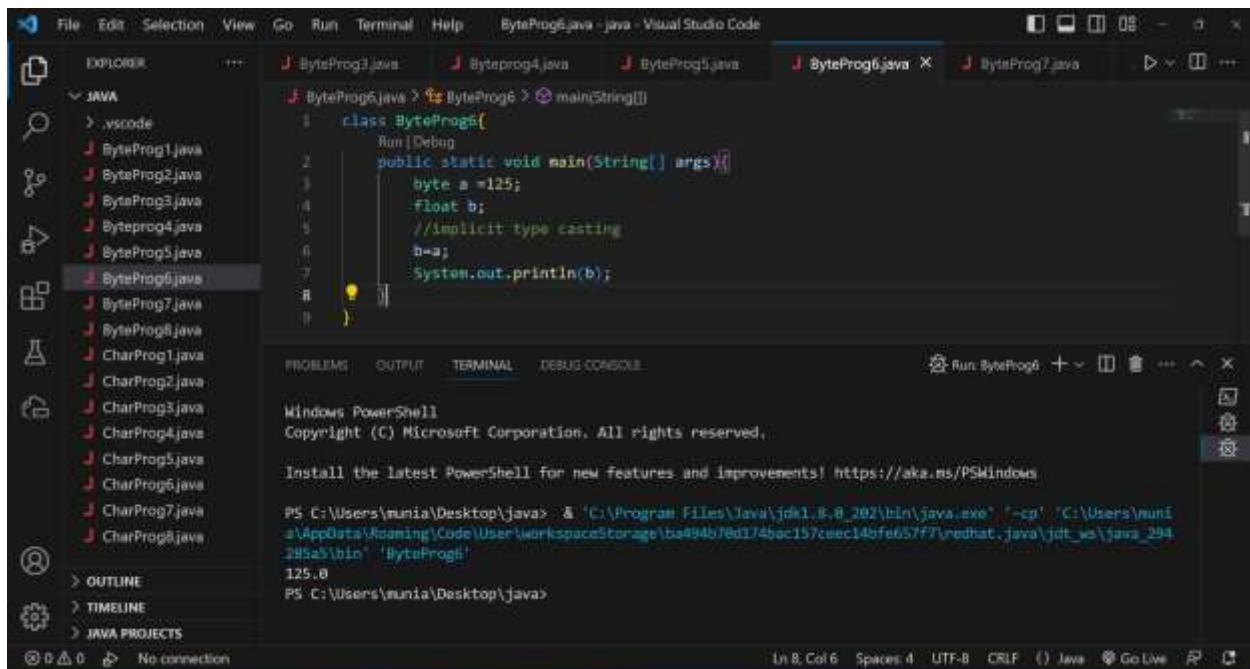
The Output pane at the bottom shows the execution of the program, displaying the output 125.

Program 14: (byte to float)

```
class ByteProg6 {  
    public static void main(String[] args) {  
        byte a = 125;  
        float b;  
        //implicit type casting  
        b = a;  
        System.out.println(b);  
    }  
}
```


Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named ByteProg6.java. The code defines a class ByteProg6 with a main method. Inside the main method, a byte variable 'a' is assigned the value 125, and a float variable 'b' is declared. The value of 'a' is assigned to 'b' without an explicit cast, demonstrating implicit type casting. The output window shows the result '125.0'.

```
class ByteProg6 {  
    public static void main(String[] args) {  
        byte a = 125;  
        float b;  
        //implicit type casting  
        b=a;  
        System.out.println(b);  
    }  
}
```

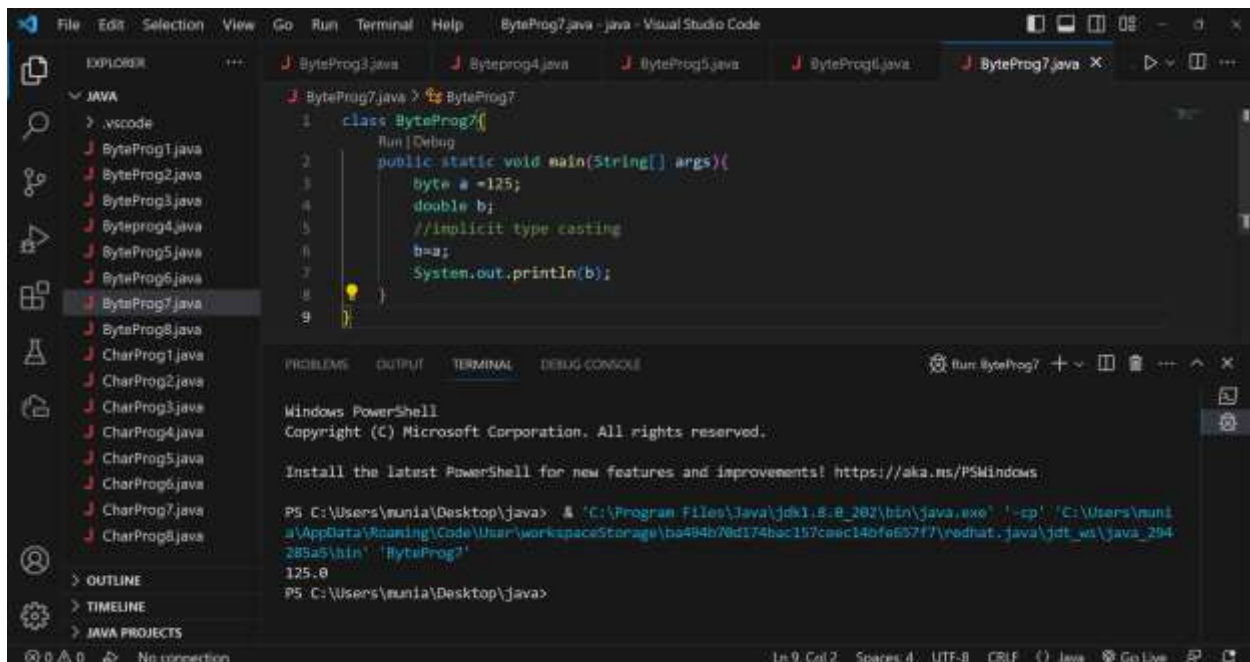
Output: 125.0

Program 15: (byte to double)

```
class ByteProg7 {  
    public static void main(String[] args) {  
        byte a = 125;  
        double b;  
        //implicit type casting  
        b=a;  
        System.out.println(b);  
    }  
}
```


Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named `ByteProg7.java`. The code defines a class `ByteProg7` with a `main` method. Inside the `main` method, a `byte` variable `a` is initialized to 125, and a `double` variable `b` is declared. The code then performs an implicit type casting from `byte` to `double` by assigning `a` to `b` (`b=a;`), followed by printing `b` using `System.out.println(b);`. The terminal output shows the command to run the program and the resulting output `125.0`.

```
class ByteProg7{  
    public static void main(String[] args){  
        byte a =125;  
        double b;  
        //implicit type casting  
        b=a;  
        System.out.println(b);  
    }  
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

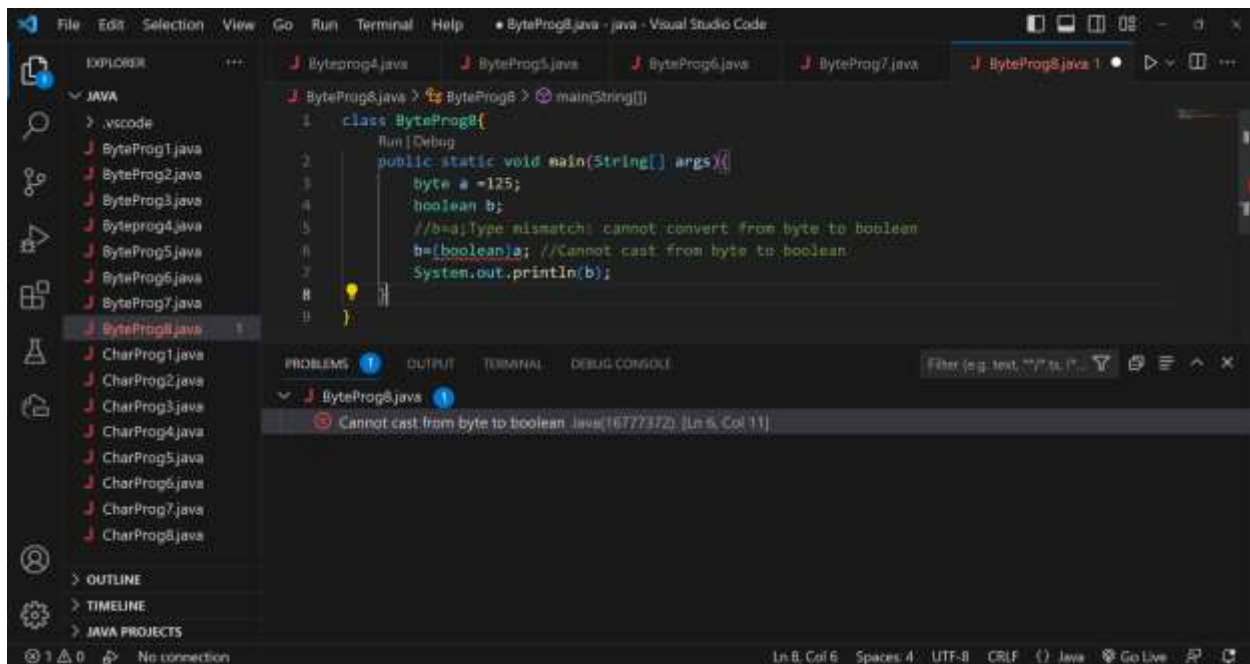
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c6ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin" "ByteProg7"
125.0
PS C:\Users\munia\Desktop\java>

Program 16: (byte to boolean)

```
class ByteProg8{  
    public static void main(String[] args){  
        byte a =125;  
        boolean b;  
        //b=a;Type mismatch: cannot convert from byte to boolean  
        b=(boolean)a; //Cannot cast from byte to boolean  
        System.out.println(b);  
    }  
}
```

Casting cannot possible

Output:

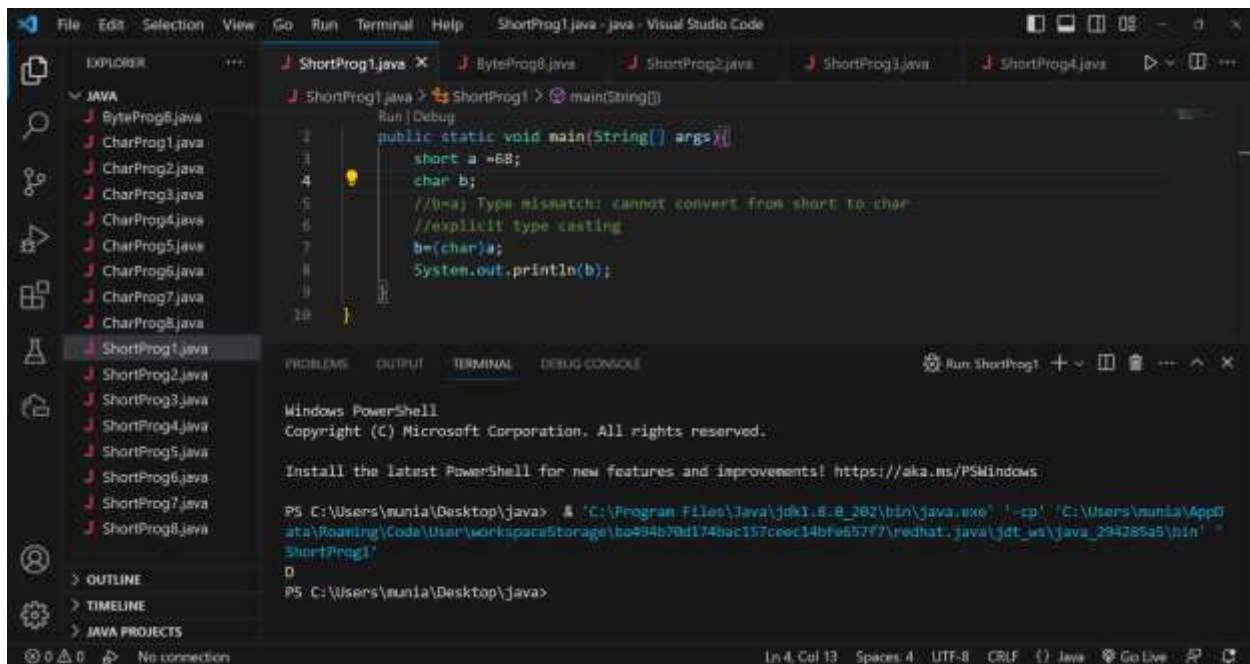


Program 17: (short to char)

```
class ShortProg1 {  
    public static void main(String[] args) {  
        short a = 68;  
        char b;  
        //b=a; Type mismatch: cannot convert from short to char  
        //explicit type casting  
        b=(char)a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including ShortProg1.java through ShortProg8.java. The main editor displays ShortProg1.java with the following code:

```
1 public class ShortProg1 {
2     public static void main(String[] args) {
3         short a = 68;
4         char b;
5         //b=a; Type mismatch: cannot convert from short to char
6         //explicit type casting
7         b=(char)a;
8         System.out.println(b);
9     }
10 }
```

A red squiggly line under the assignment `b=a;` on line 5 indicates a type mismatch error. The bottom pane shows the Output window with the command prompt output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

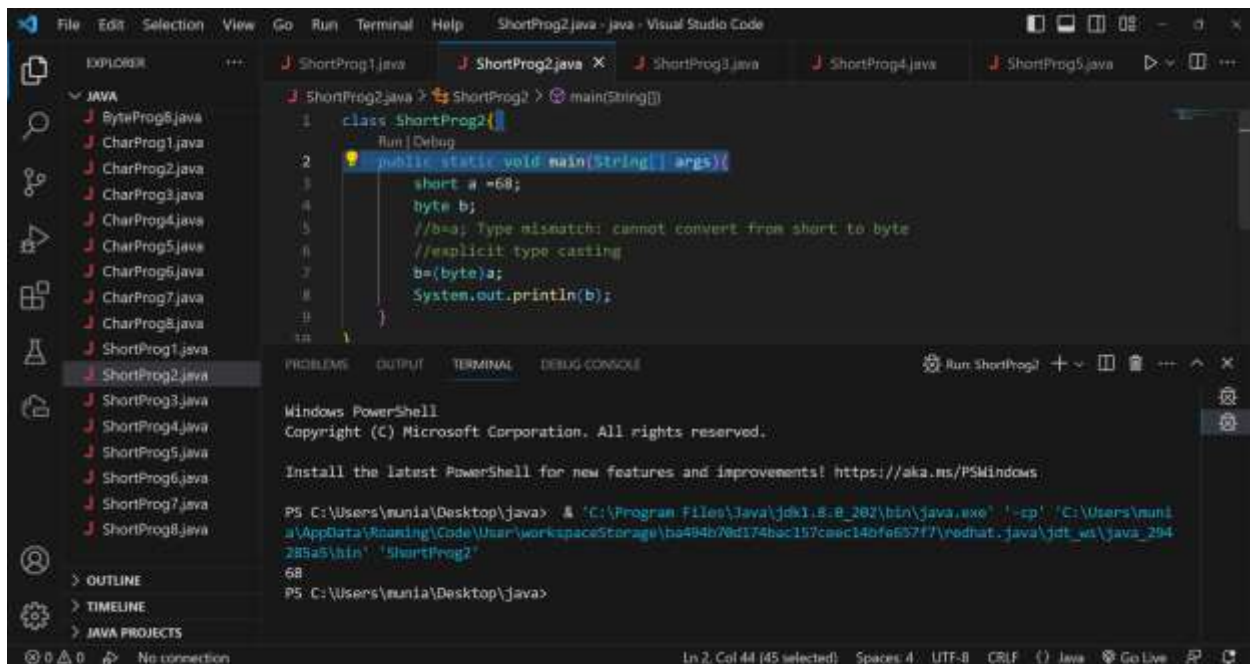
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Local\Roaming\Coda\Usar\workspacesStorage\ba494b70d174bac157ceec14bfe657f7\redhat.java\jdk_1.8\java_294285a5\bin' 'ShortProg1'
D
PS C:\Users\munia\Desktop\java>
```

Program 18: (short to byte)

```
class ShortProg2{
    public static void main(String[] args){
        short a=68;
        byte b;
        //b=a; Type mismatch: cannot convert from short to byte
        //explicit type casting
        b=(byte)a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named `ShortProg2.java`. The code defines a class `ShortProg2` with a `main` method. Inside the `main` method, a `short` variable `a` is initialized with the value 68. A `byte` variable `b` is declared. A comment indicates a type mismatch: `//b=a; Type mismatch: cannot convert from short to byte`. To resolve this, explicit type casting is used: `b=(byte)a;`. Finally, `System.out.println(b);` is called. The output window at the bottom shows the command prompt execution: `PS C:\Users\munia\Desktop\java> java -cp 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin' 'ShortProg2'`, followed by the output `68`.

```
class ShortProg2{
    public static void main(String[] args){
        short a =68;
        byte b;
        //b=a; Type mismatch: cannot convert from short to byte
        //explicit type casting
        b=(byte)a;
        System.out.println(b);
    }
}
```

```
PS C:\Users\munia\Desktop\java> java -cp 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin' 'ShortProg2'
68
```

Program 19: (short to short)

class ShortProg3{

public static void main(String[] args){

short a=152;

short b;

//casting not required

b=a;

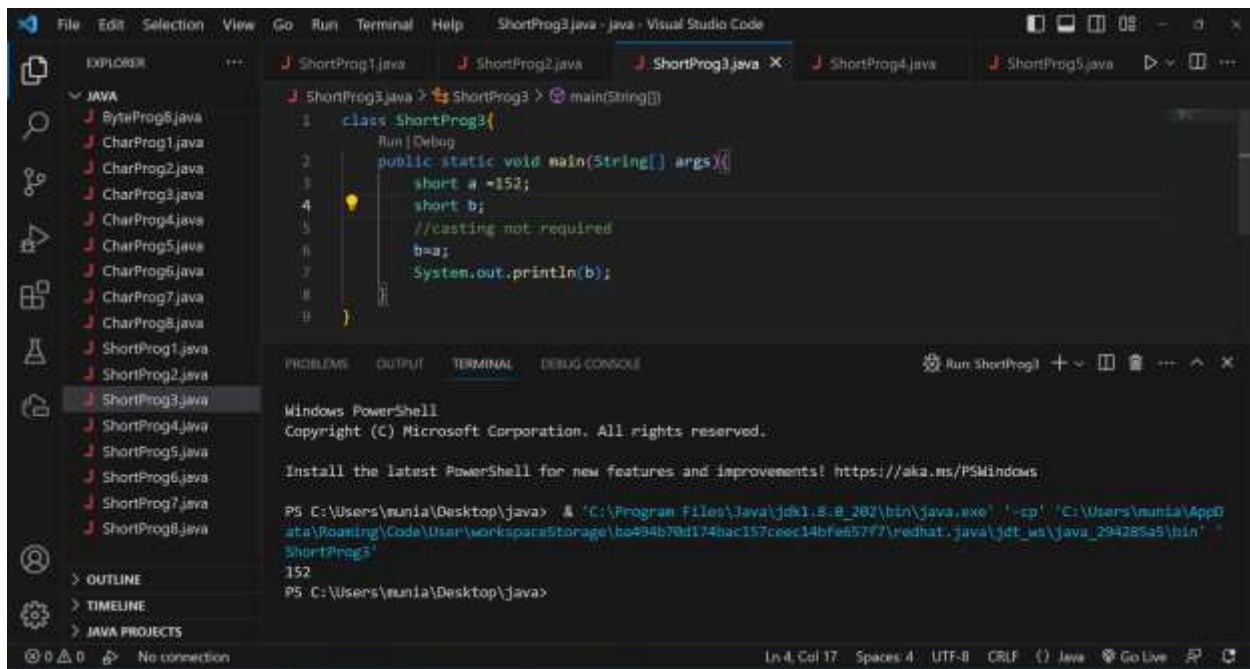
System.out.println(b);

}

}

Casting not required

Output:



```
class ShortProg3{
    public static void main(String[] args){
        short a = 152;
        short b;
        //casting not required
        b=a;
        System.out.println(b);
    }
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Coda\User\workspacesStorage\ba494b70d174bac157ceec14bfe657f7\redhat.java\jdk_18\java_294285a5\bin' "ShortProg3"

152

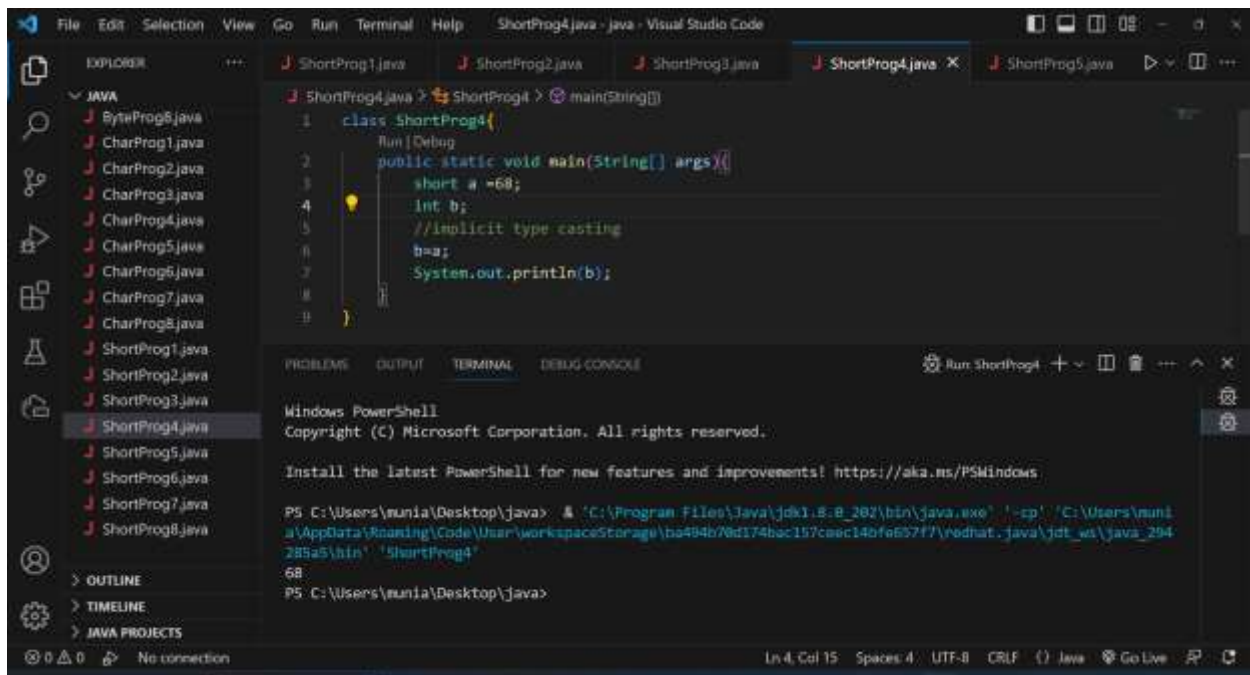
PS C:\Users\munia\Desktop\java>

Program 20: (short to int)

```
class ShortProg4{
    public static void main(String[] args){
        short a = 68;
        int b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named `ShortProg4.java`. The code defines a class `ShortProg4` with a `main` method. Inside the `main` method, a `short` variable `a` is initialized with the value 68, and an `int` variable `b` is declared. A comment `//implicit type casting` is present above the assignment `b=a;`. The `System.out.println(b);` statement prints the value of `b`. The terminal at the bottom shows the command to run the program and the output `68`.

```
class ShortProg4{
    public static void main(String[] args){
        short a =68;
        int b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

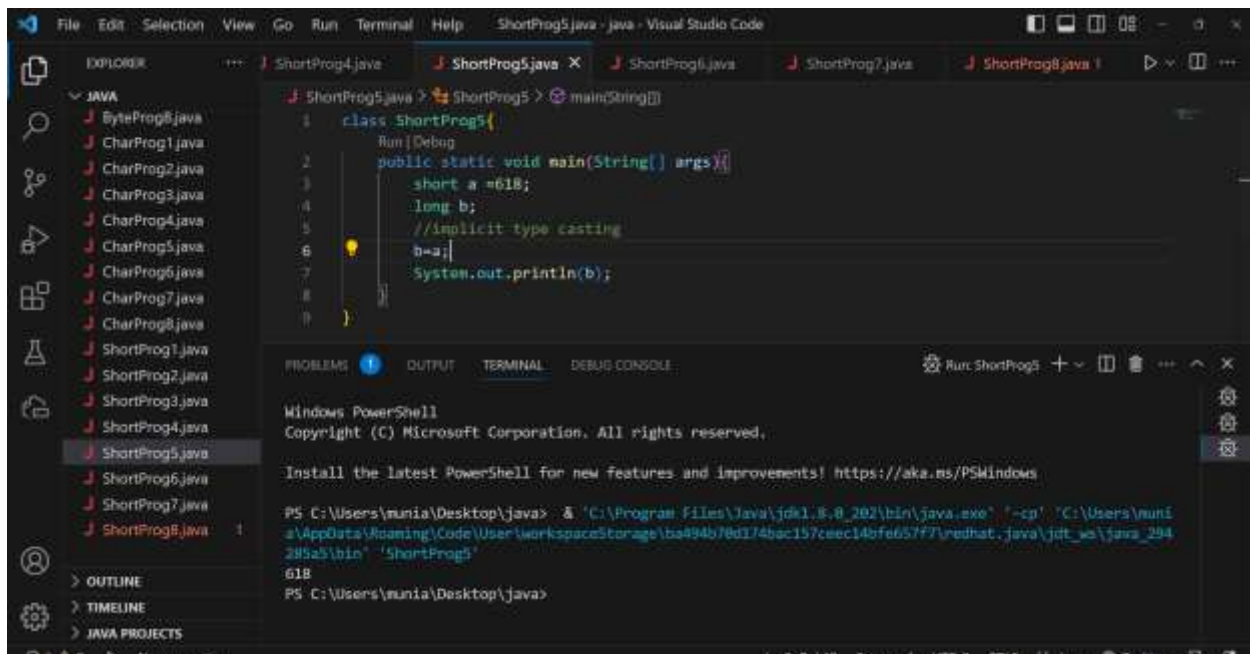
```
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin" "ShortProg4"
68
PS C:\Users\munia\Desktop\java>
```

Program 21: (short to long)

```
class ShortProg5{
    public static void main(String[] args){
        short a =618;
        long b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, with 'ShortProg5.java' selected. The main editor displays the code for 'ShortProg5.java':

```
1 class ShortProg5 {  
2     public static void main(String[] args) {  
3         short a = 618;  
4         long b;  
5         //implicit type casting  
6         b=a;  
7         System.out.println(b);  
8     }  
9 }
```

The 'TERMINAL' pane at the bottom shows the command prompt output:

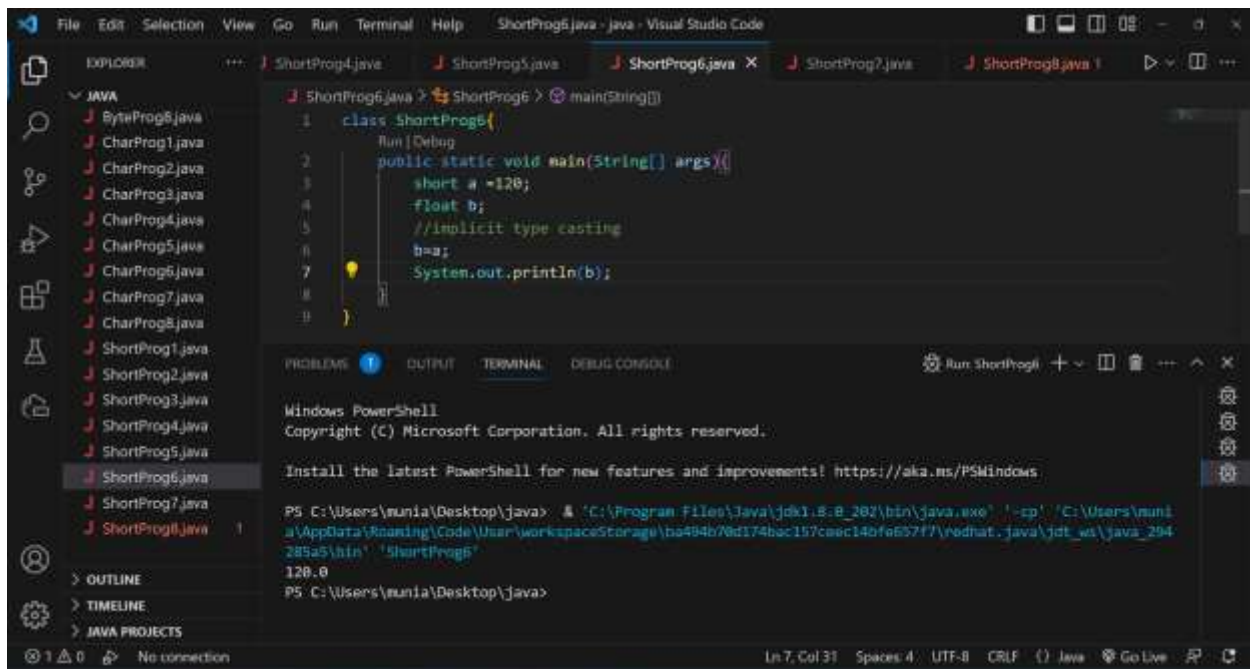
```
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' "-cp" 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157ccec14bfe657f7\redhat.java\jdt_ws\java_294285a3\bin' 'ShortProg5'  
618  
PS C:\Users\munia\Desktop\java>
```

Program 22: (short to float)

```
class ShortProg6 {  
    public static void main(String[] args) {  
        short a = 120;  
        float b;  
        //implicit type casting  
        b = a;  
        System.out.println(b);  
    }  
}
```


Implicit type casting

Output:



```
class ShortProg6 {
    public static void main(String[] args) {
        short a = 120;
        float b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin" "ShortProg6"

120.0

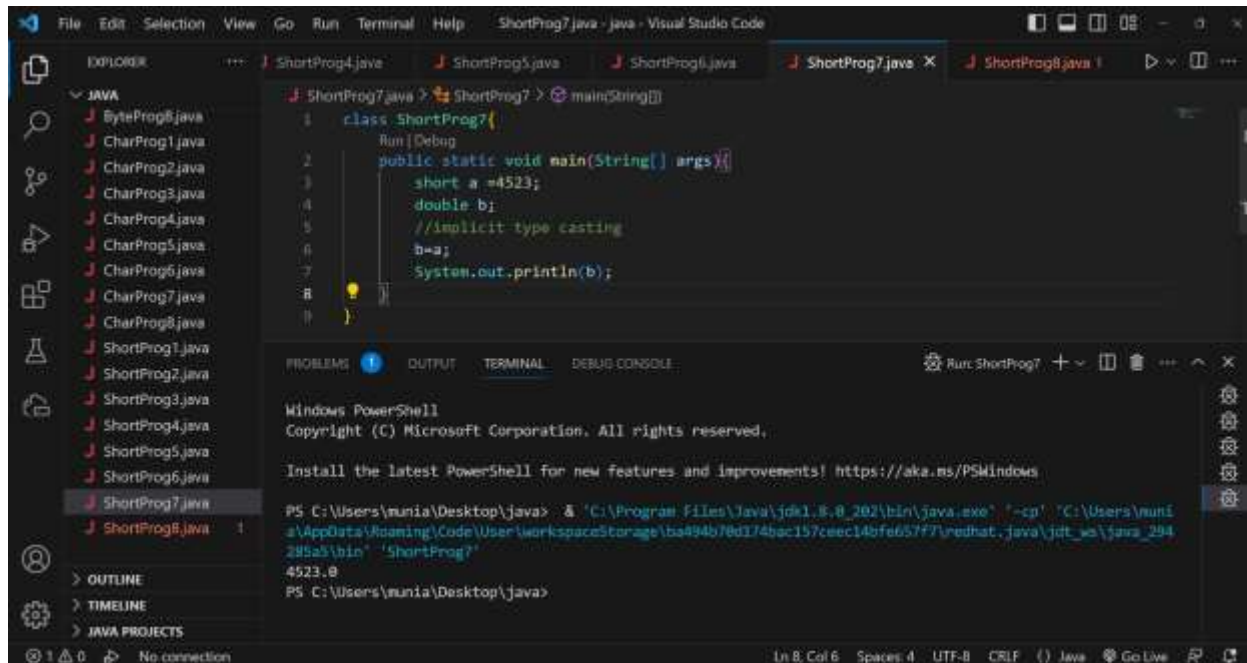
PS C:\Users\munia\Desktop\java>

Program 23: (short to double)

```
class ShortProg7 {
    public static void main(String[] args) {
        short a = 4523;
        double b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```


Implicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, with 'ShortProg7.java' selected. The main editor displays the code for 'ShortProg7.java', which defines a class 'ShortProg7' with a 'main' method. The code includes a 'short' variable 'a' with the value 4523, a 'double' variable 'b', and an assignment 'b=a;' with a comment '//implicit type casting'. The 'main' method prints the value of 'b'. The Output pane at the bottom shows the execution results, displaying '4523.0'. The terminal pane shows the command used to run the program: 'PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d374bac157ceec14bfe657f7\redhat.java\jdt_ws\java_294285a5\bin' 'ShortProg7''.

```
class ShortProg7{
    public static void main(String[] args){
        short a =4523;
        double b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

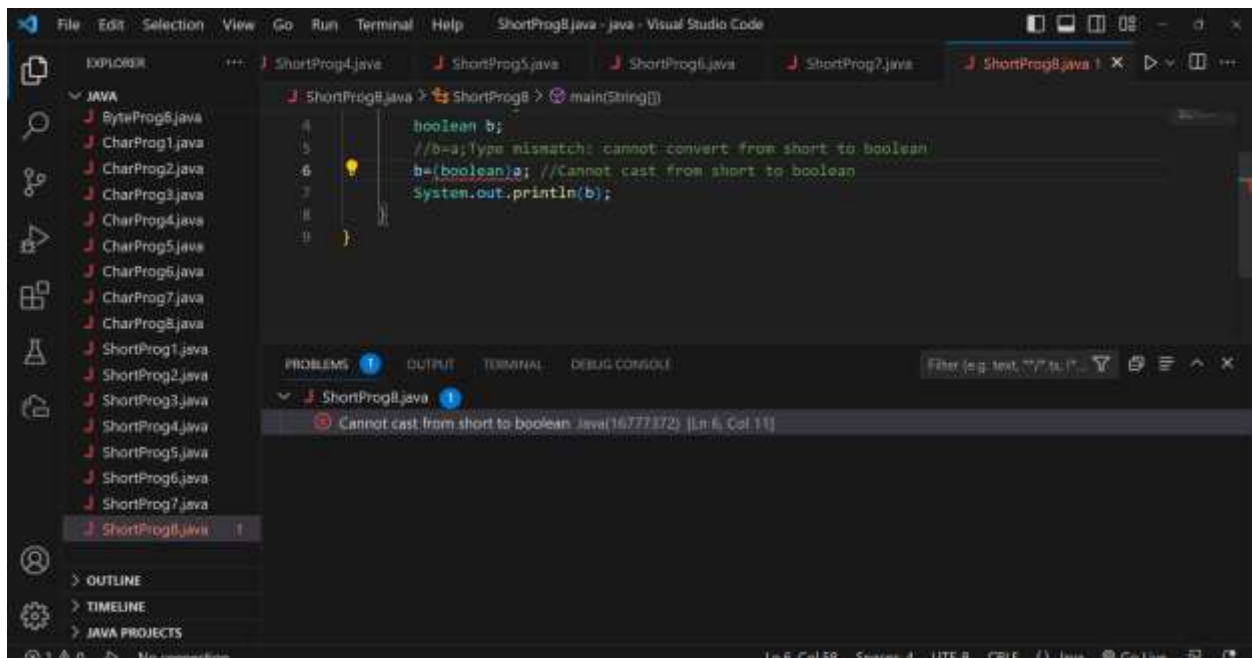
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d374bac157ceec14bfe657f7\redhat.java\jdt_ws\java_294285a5\bin' 'ShortProg7'
4523.0
PS C:\Users\munia\Desktop\java>

Program 24: (short to boolean)

```
class ShortProg8{
    public static void main(String[] args){
        short a=68;
        boolean b;
        //b=a;Type mismatch: cannot convert from short to boolean
        b=(boolean)a; //Cannot cast from short to boolean
        System.out.println(b);
    }
}
```

Casting cannot possible

Output:

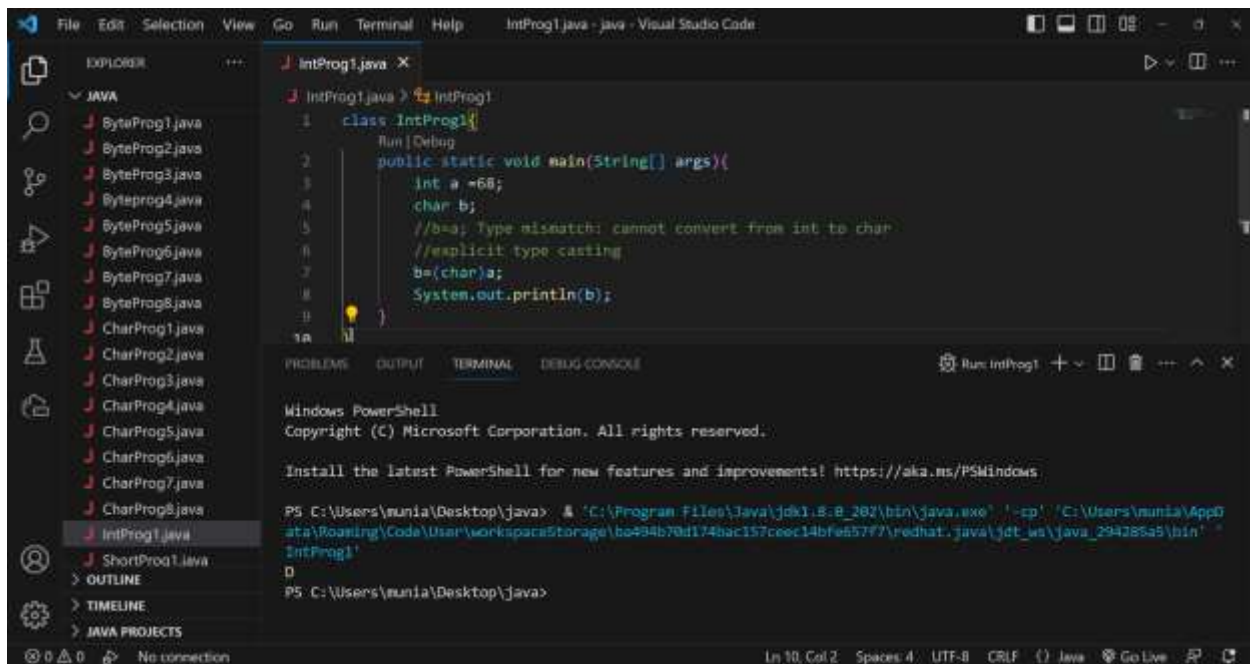


Program 25: (int to char)

```
class IntProg1 {
    public static void main(String[] args){
        int a =68;
        char b;
        //b=a; Type mismatch: cannot convert from int to char
        //explicit type casting
        b=(char)a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a file named `IntProg1.java` open. The code defines a class `IntProg1` with a `main` method. Inside the `main` method, an integer `a` is assigned the value 68. A character `b` is declared, and a comment indicates a type mismatch: `//b=a; Type mismatch: cannot convert from int to char`. The code then uses explicit type casting: `b=(char)a;`, followed by `System.out.println(b);`. The output window at the bottom shows the command prompt running the program, which prints the character `D` (the ASCII character for 68).

```
1 class IntProg1 {
2     public static void main(String[] args) {
3         int a = 68;
4         char b;
5         //b=a; Type mismatch: cannot convert from int to char
6         //explicit type casting
7         b=(char)a;
8         System.out.println(b);
9     }
10 }
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Local\Roaming\Coda\User\workspaceStorage\ba494b70d174bac157ceec14bfe657f7\redhat.java\jdk_ws\java_294285a5\bin' "IntProg1"

D

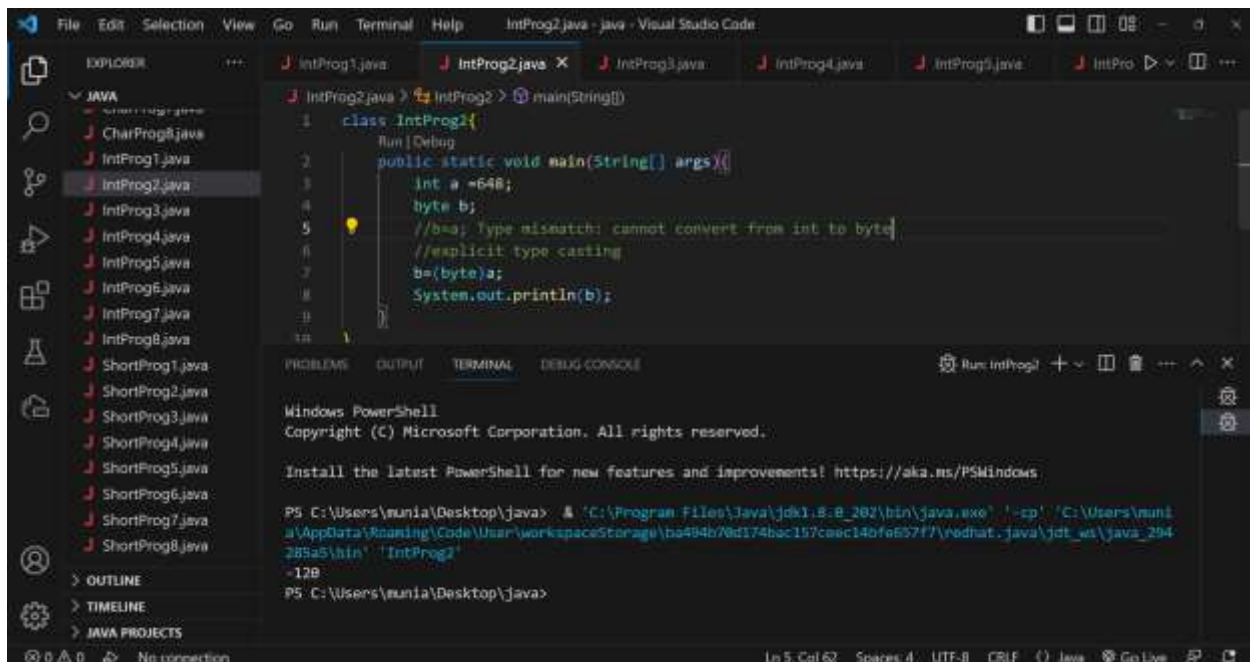
PS C:\Users\munia\Desktop\java>

Program 26: (int to byte)

```
class IntProg2 {
    public static void main(String[] args) {
        int a = 648;
        byte b;
        //b=a; Type mismatch: cannot convert from int to byte
        //explicit type casting
        b=(byte)a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named `IntProg2.java`. The code is as follows:

```
1 class IntProg2{
2     public static void main(String[] args){
3         int a = 648;
4         byte b;
5         //b=a; Type mismatch: cannot convert from int to byte
6         //explicit type casting
7         b=(byte)a;
8         System.out.println(b);
9     }
10 }
```

A red squiggly line under the comment on line 5 indicates a type mismatch error. The output window at the bottom shows the command prompt output:

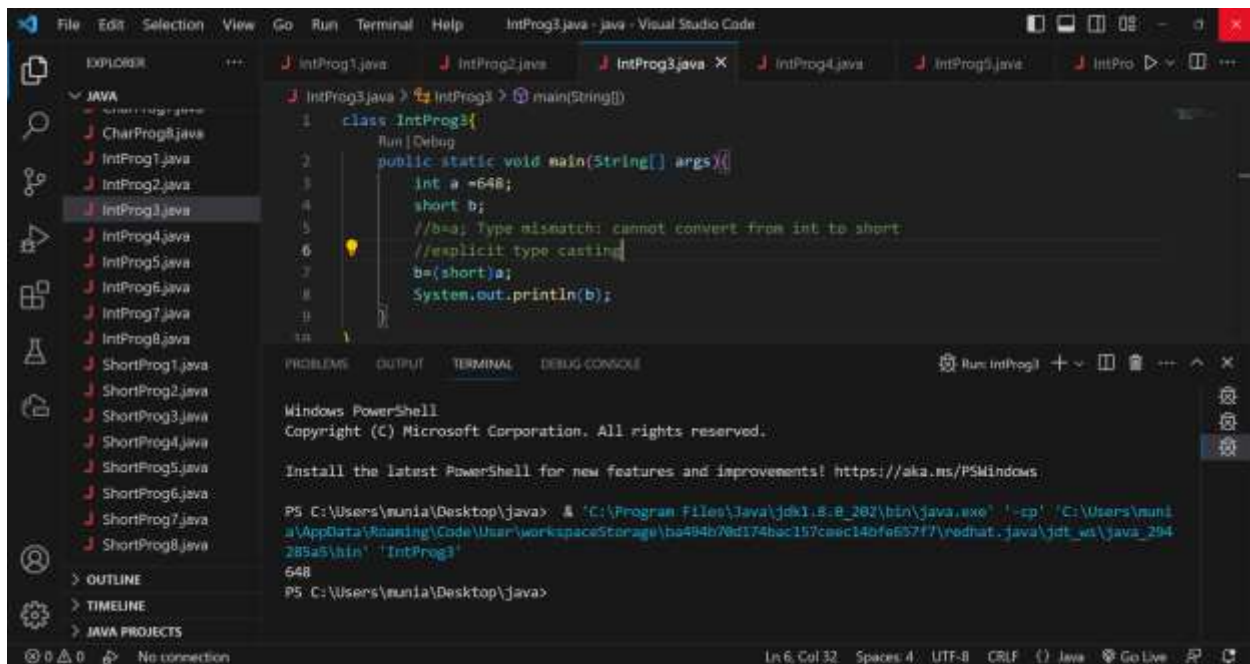
```
PS C:\Users\munia\Desktop\java> java -cp 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin' 'IntProg2'
-128
```

Program 27: (int to short)

```
class IntProg3{
    public static void main(String[] args){
        int a = 648;
        short b;
        //b=a; Type mismatch: cannot convert from int to short
        //explicit type casting
        b=(short)a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named `IntProg3.java`. The code defines a class `IntProg3` with a `main` method. Inside the `main` method, an `int` variable `a` is assigned the value 648, and a `short` variable `b` is declared. A comment indicates a type mismatch: `//b=a; Type mismatch: cannot convert from int to short`. The next line shows explicit type casting: `b=(short)a;`. Finally, `System.out.println(b);` is used to print the value of `b`. The output window at the bottom shows the command prompt running the program, which outputs the value 648.

```
1 class IntProg3{
2     public static void main(String[] args){
3         int a =648;
4         short b;
5         //b=a; Type mismatch: cannot convert from int to short
6         //explicit type casting
7         b=(short)a;
8         System.out.println(b);
9     }
10 }
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin' 'IntProg3'

648

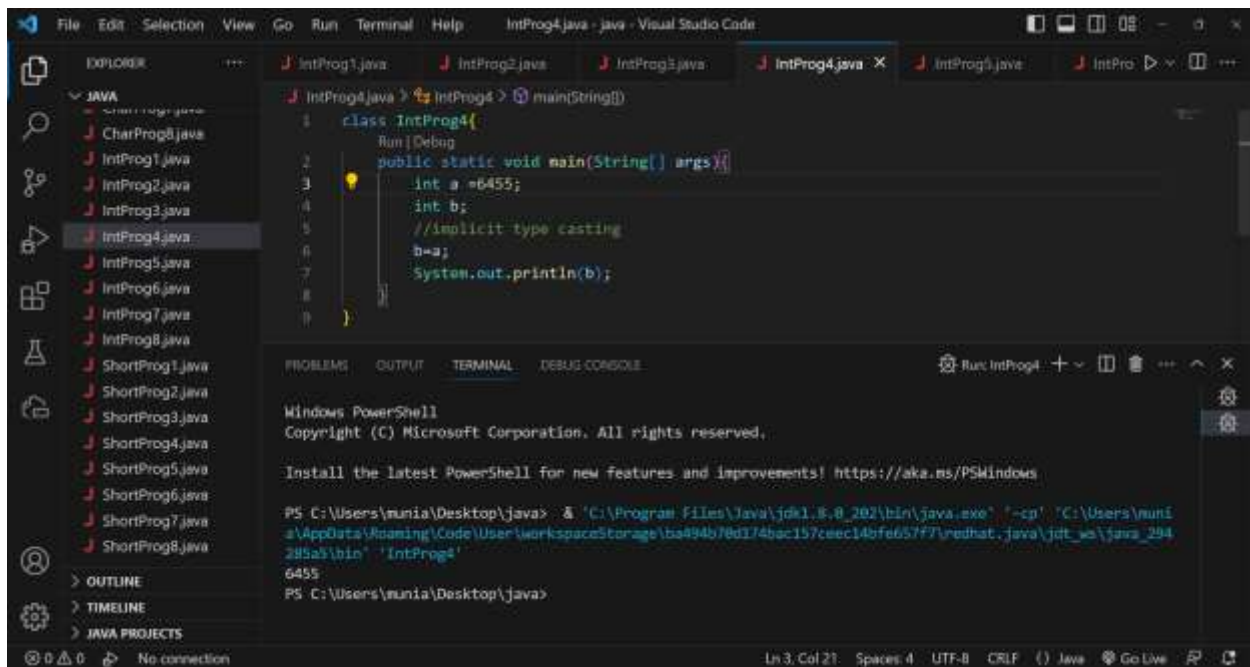
PS C:\Users\munia\Desktop\java>

Program 28: (int to int)

```
class IntProg3 {
    public static void main(String[] args){
        int a=6455;
        int b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

Casting not required

Output:



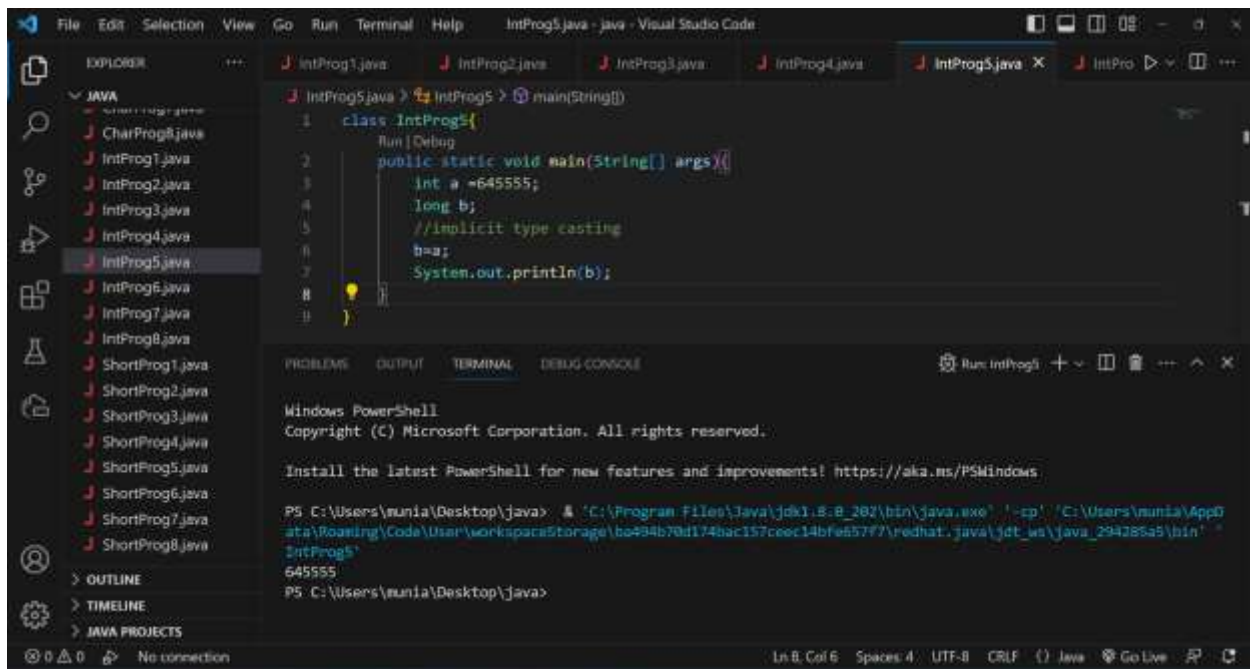
```
File Edit Selection View Go Run Terminal Help IntProg4.java - java - Visual Studio Code
EXPLORER
  JAVA
    CharProg.java
    IntProg1.java
    IntProg2.java
    IntProg3.java
    IntProg4.java
    IntProg5.java
    IntProg6.java
    IntProg7.java
    IntProg8.java
    ShortProg1.java
    ShortProg2.java
    ShortProg3.java
    ShortProg4.java
    ShortProg5.java
    ShortProg6.java
    ShortProg7.java
    ShortProg8.java
  OUTLINE
  TIMELINE
  JAVA PROJECTS
  PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
  Run: IntProg4
  Windows PowerShell
  Copyright (C) Microsoft Corporation. All rights reserved.
  Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
  PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba894b7ed174bac157ceec14bfe657f7\redhat.java\jdt_ws\java_294285a3\bin' 'IntProg4'
  6455
  PS C:\Users\munia\Desktop\java>
```

Program 29: (int to long)

```
class IntProg5 {
    public static void main(String[] args) {
        int a = 645555;
        long b;
        //implicit type casting
        b = a;
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



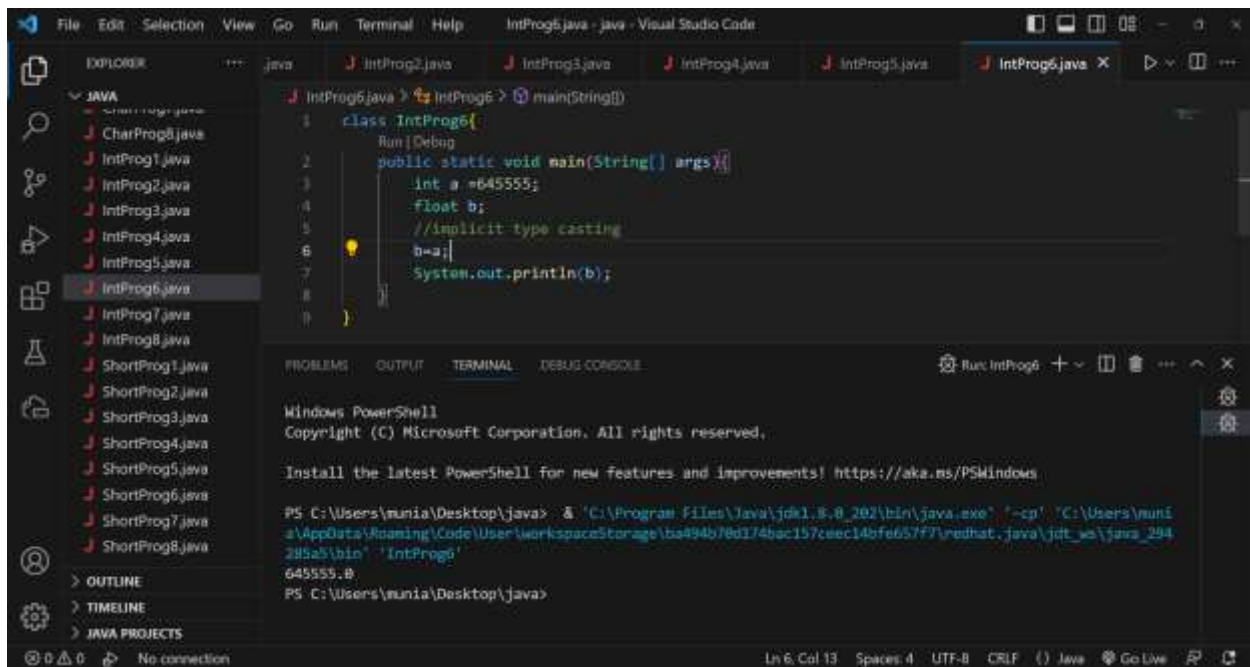
```
File Edit Selection View Go Run Terminal Help IntProg5.java - java - Visual Studio Code
EXPLORER
  JAVA
    IntProg1.java
    IntProg2.java
    IntProg3.java
    IntProg4.java
    IntProg5.java
    IntProg6.java
    IntProg7.java
    IntProg8.java
    ShortProg1.java
    ShortProg2.java
    ShortProg3.java
    ShortProg4.java
    ShortProg5.java
    ShortProg6.java
    ShortProg7.java
    ShortProg8.java
  OUTLINE
  TIMELINE
  JAVA PROJECTS
  PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
  Run IntProg5
  Windows PowerShell
  Copyright (C) Microsoft Corporation. All rights reserved.
  Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
  PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Local\Roaming\Coda\User\workspaceStorage\ba94b70d174bac157ceec14bfe657f7\redhat.java\jdk_ws\java_294285a5\bin' 'IntProg5'
  645555
  PS C:\Users\munia\Desktop\java>
```

Program 30: (int to float)

```
class IntProg6{
    public static void main(String[] args){
        int a=645555;
        float b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```


Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named `IntProg6.java`. The code defines a class `IntProg6` with a `main` method. Inside the `main` method, an `int` variable `a` is assigned the value `645555`, and a `float` variable `b` is assigned the value of `a`. A comment `//implicit type casting` is placed above the assignment `b=a;`. The program then prints the value of `b` using `System.out.println(b);`. The output window at the bottom shows the command prompt execution, which results in `645555.0`.

```
class IntProg6{
    public static void main(String[] args){
        int a =645555;
        float b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

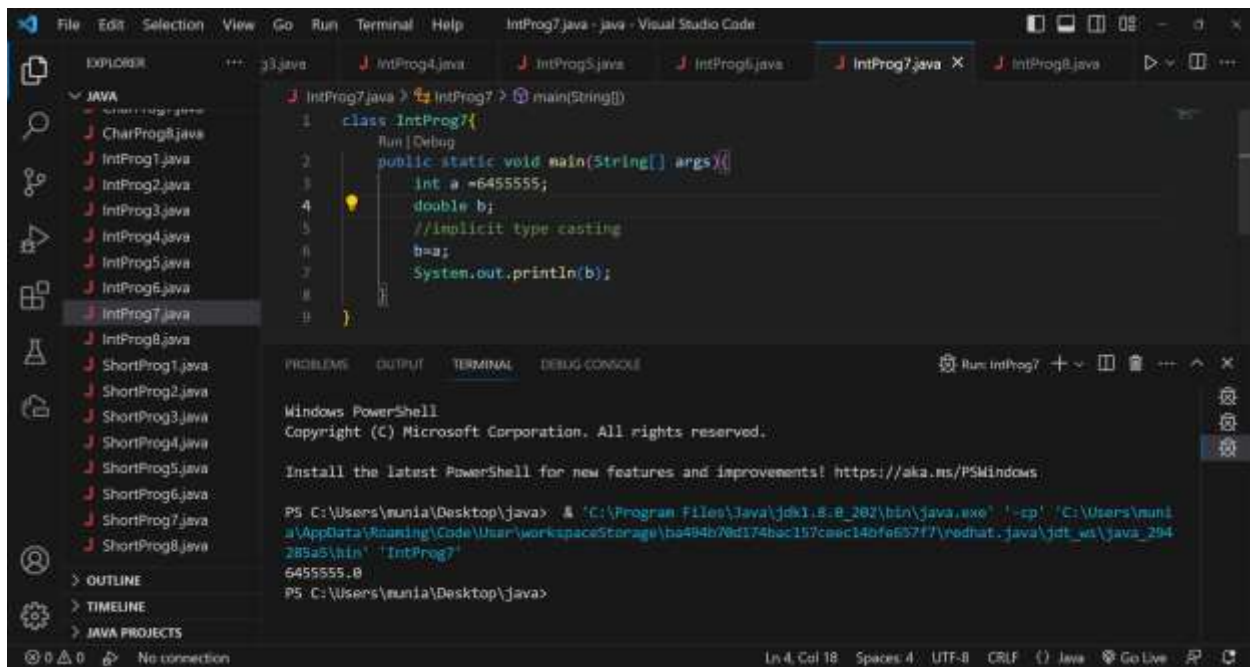
```
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' "-cp" 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b7ed174bac157ceec14bfe657f7\redhat.java\jdt_ws\java_29428525\bin' 'IntProg6'
645555.0
```

Program 31: (int to double)

```
class IntProg7{
    public static void main(String[] args){
        int a =6455555;
        double b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```


Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named `IntProg7.java`. The code defines a class `IntProg7` with a `main` method. Inside the `main` method, an `int` variable `a` is assigned the value `6455555`, and a `double` variable `b` is assigned the value of `a`. A comment `//implicit type casting` is placed above the assignment `b=a;`. The `System.out.println(b);` statement prints the value of `b`. The output window at the bottom shows the command prompt running the Java program, which outputs `6455555.0`.

```
class IntProg7{  
    public static void main(String[] args){  
        int a =6455555;  
        double b;  
        //implicit type casting  
        b=a;  
        System.out.println(b);  
    }  
}
```

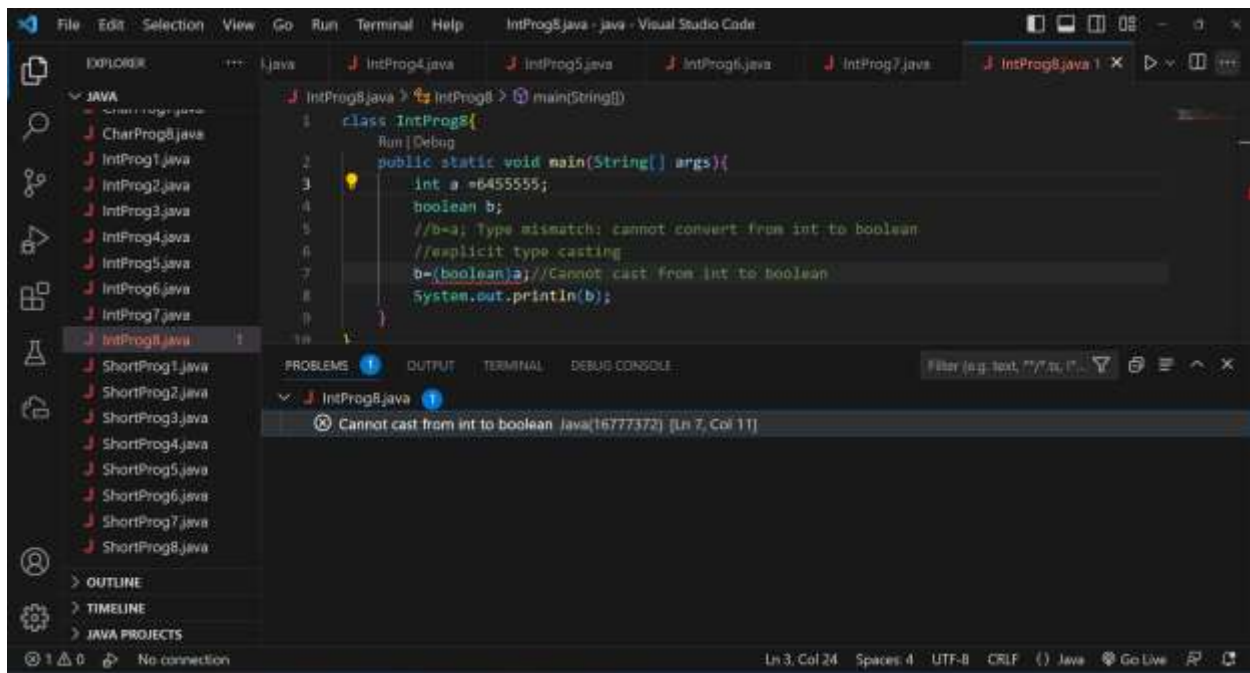
```
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdk_wa\java_294285a5\bin" "IntProg7"  
6455555.0  
PS C:\Users\munia\Desktop\java>
```

Program 32: (int to boolean)

```
class IntProg8{  
    public static void main(String[] args){  
        int a=6455555;  
        boolean b;  
        //b=a; Type mismatch: cannot convert from int to boolean  
        //explicit type casting  
        b=(boolean)a;//Cannot cast from int to boolean  
        System.out.println(b);  
    }  
}
```

Casting cannot possible

Output:

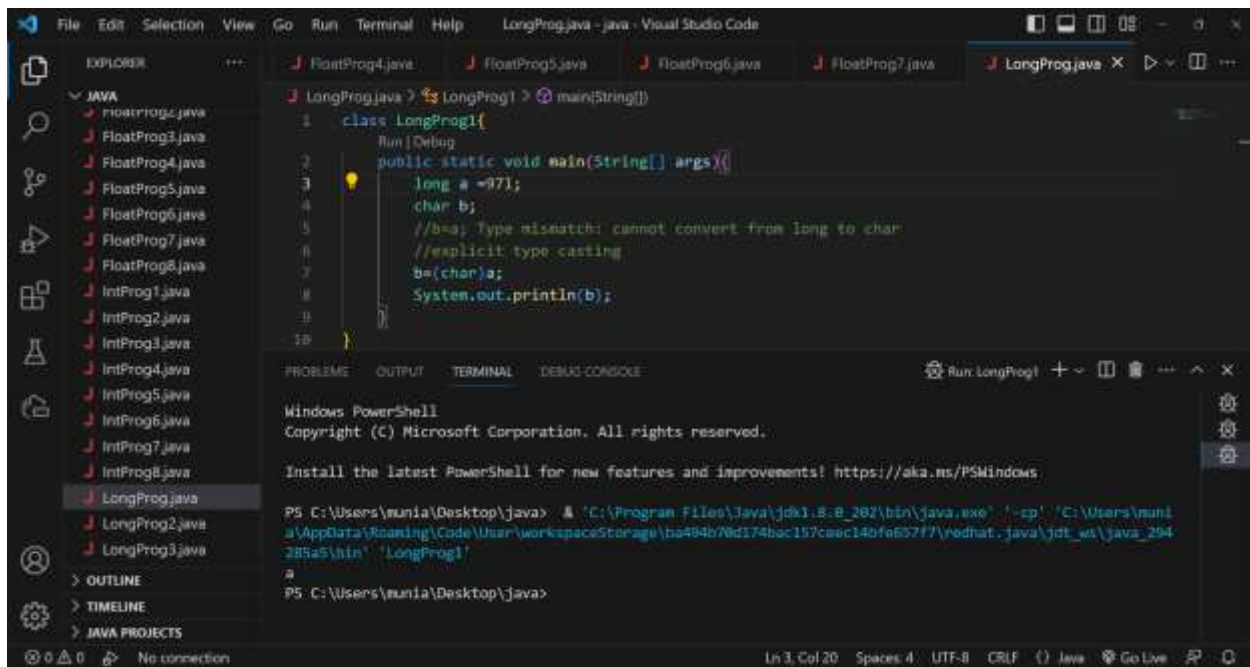


Program 33: (long to char)

```
class LongProg1 {  
    public static void main(String[] args) {  
        long a = 97l;  
        char b;  
        //b=a; Type mismatch: cannot convert from long to char  
        //explicit type casting  
        b=(char)a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including LongProg1.java through LongProg8.java. The main editor displays LongProg1.java with the following code:

```
1 class LongProg1{
2     public static void main(String[] args){
3         long a = 971;
4         char b;
5         //b=a; Type mismatch: cannot convert from long to char
6         //explicit type casting
7         b=(char)a;
8         System.out.println(b);
9     }
10 }
```

A red squiggly line under the assignment `b=a;` indicates a type mismatch error. The bottom panel shows the OUTPUT window with the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

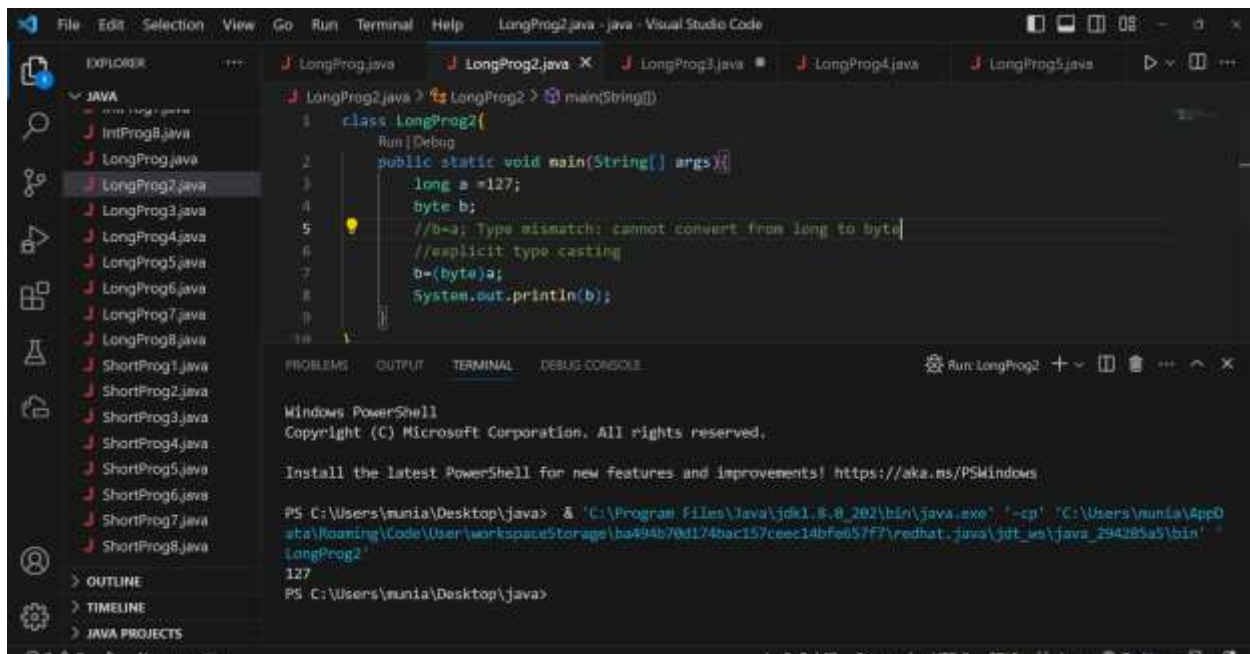
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c8ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin" "LongProg1"
a
PS C:\Users\munia\Desktop\java>
```

Program 34: (long to byte)

```
class LongProg2 {
    public static void main(String[] args){
        long a = 1271;
        byte b;
        //b=a; Type mismatch: cannot convert from long to char
        //explicit type casting
        b=(byte)a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including LongProg2.java. The main editor displays the code for LongProg2.java, which defines a class LongProg2 with a main method. The code includes a long variable 'a' with the value 127, a byte variable 'b', and an attempt to assign 'a' to 'b' without explicit casting, which has triggered a compiler error: '//b=a; Type mismatch: cannot convert from long to byte'. Below this, the code uses explicit casting: 'b=(byte)a;'. The output pane at the bottom shows the result of running the program: '127'.

```
class LongProg2 {  
    public static void main(String[] args) {  
        long a = 127;  
        byte b;  
        //b=a; Type mismatch: cannot convert from long to byte  
        //explicit type casting  
        b=(byte)a;  
        System.out.println(b);  
    }  
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

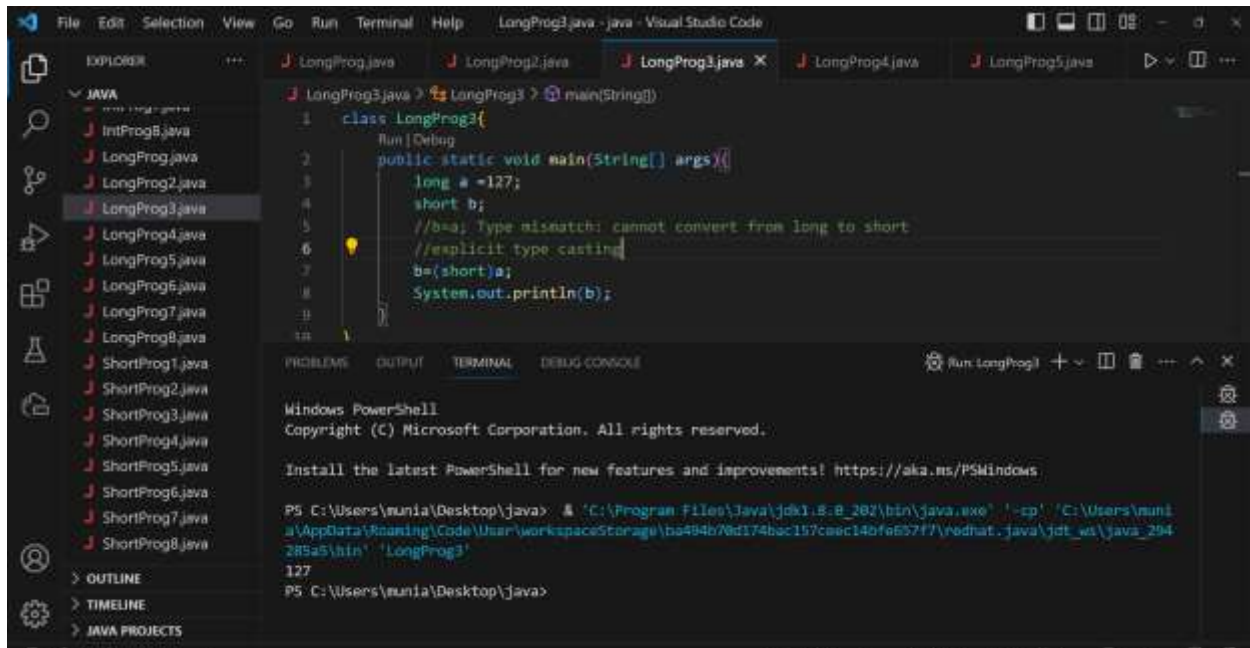
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\b4694b7bd174bac157ceec14bfe657f7\redhat_java\jdt_ws\java_294285a5\bin" "LongProg2"
127
PS C:\Users\munia\Desktop\java>

Program 35: (long to short)

```
class LongProg3 {  
    public static void main(String[] args) {  
        long a = 127L;  
        short b;  
        //b=a; Type mismatch: cannot convert from long to short  
        //explicit type casting  
        b=(short)a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named LongProg3.java. The code defines a class LongProg3 with a main method. Inside the main method, a long variable 'a' is assigned the value 127. A short variable 'b' is declared. A comment indicates a type mismatch: '//b=a; Type mismatch: cannot convert from long to short'. The code then uses explicit type casting: '//explicit type casting' followed by 'b=(short)a;'. Finally, 'System.out.println(b);' is called. The terminal at the bottom shows the command to run the program, and the output is '127'.

```
class LongProg3{
    public static void main(String[] args){
        long a =127;
        short b;
        //b=a; Type mismatch: cannot convert from long to short
        //explicit type casting
        b=(short)a;
        System.out.println(b);
    }
}
```

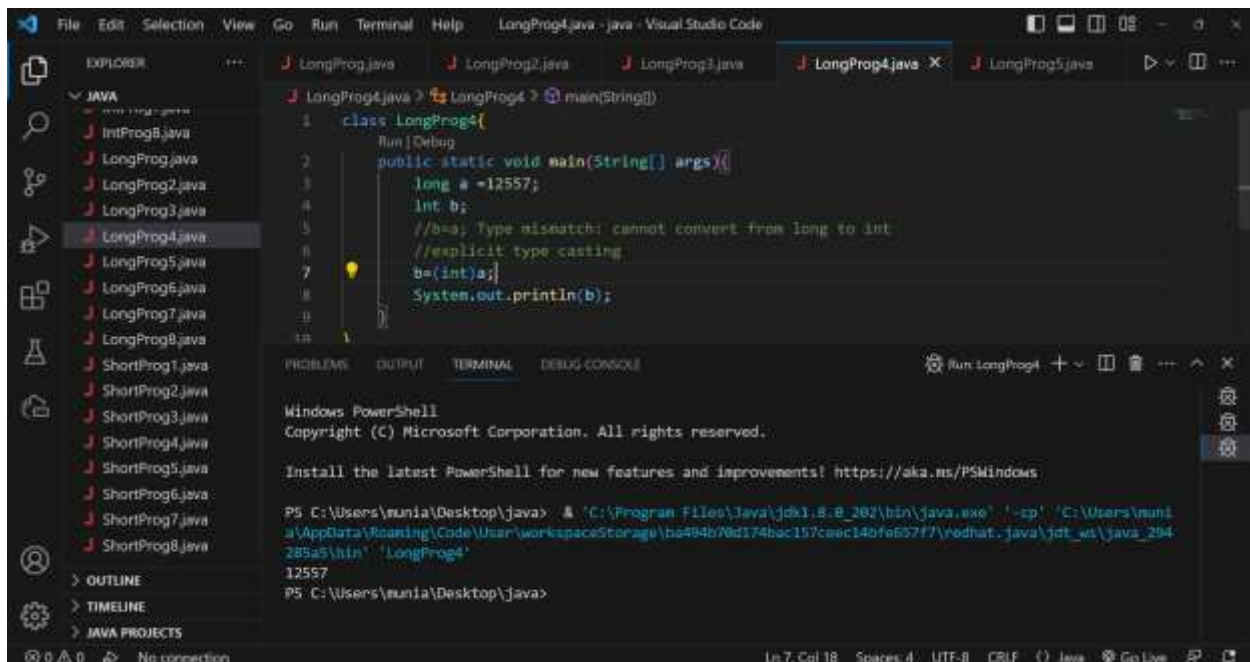
```
PS C:\Users\munia\Desktop\java> java -cp 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c9ec140fe857f7\redhat.java\jdt_ws\java_294285a5\bin' 'LongProg3'
127
PS C:\Users\munia\Desktop\java>
```

Program 36: (long to int)

```
class LongProg4{
    public static void main(String[] args){
        long a =125571;
        int b;
        //b=a; Type mismatch: cannot convert from long to int
        //explicit type casting
        b=(int)a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including LongProg4.java. The main editor displays the code for LongProg4.java, which defines a class LongProg4 with a main method. The code includes a long variable 'a' with the value 12557, an int variable 'b', and an explicit type cast from long to int: `b=(int)a;`. A comment above the cast reads: `//explicit type casting`. The main method prints the value of 'b'. The Output pane at the bottom shows the execution results in a Windows PowerShell terminal, displaying the number 12557.

```
class LongProg4{
    public static void main(String[] args){
        long a =12557;
        int b;
        //b=a; Type mismatch: cannot convert from long to int
        //explicit type casting
        b=(int)a;
        System.out.println(b);
    }
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin" "LongProg4"

12557

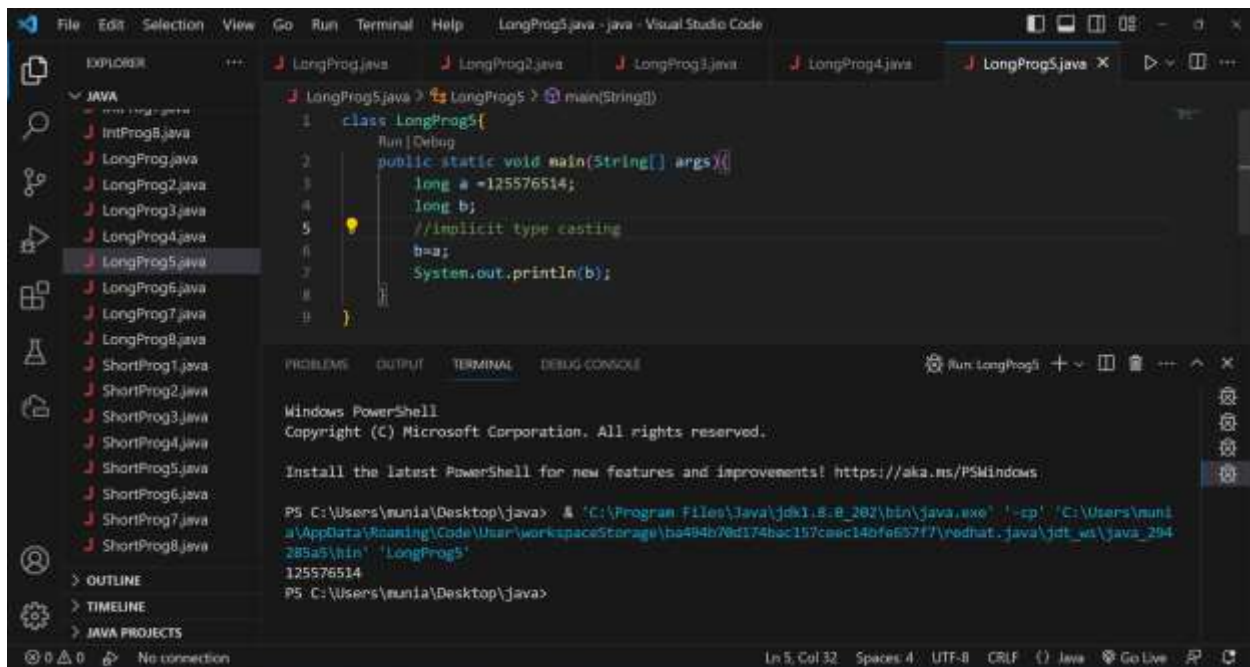
PS C:\Users\munia\Desktop\java>

Program 37: (long to long)

```
class LongProg5{
    public static void main(String[] args){
        long a =125576514l;
        long b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```


Casting not required

Output:



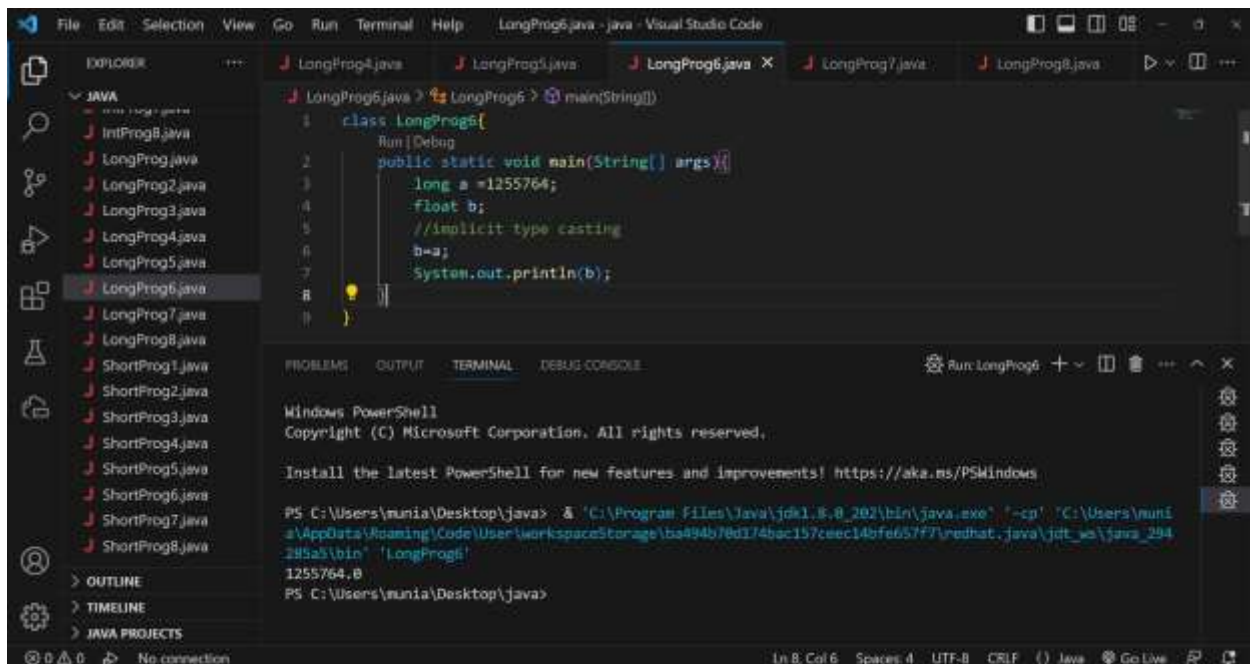
```
File Edit Selection View Go Run Terminal Help LongProg5.java - java - Visual Studio Code
EXPLORER
  JAVA
    LongProg1.java
    LongProg2.java
    LongProg3.java
    LongProg4.java
    LongProg5.java
    LongProg6.java
    LongProg7.java
    LongProg8.java
    ShortProg1.java
    ShortProg2.java
    ShortProg3.java
    ShortProg4.java
    ShortProg5.java
    ShortProg6.java
    ShortProg7.java
    ShortProg8.java
  OUTLINE
  TIMELINE
  JAVA PROJECTS
  PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
  Run longProg5
  Windows PowerShell
  Copyright (C) Microsoft Corporation. All rights reserved.
  Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
  PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c0ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin" "LongProg5"
  125576514
  PS C:\Users\munia\Desktop\java>
```

Program 38: (long to float)

```
class LongProg6{
    public static void main(String[] args){
        long a =12557641;
        float b;
        //implicit type casting
        b=a;
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java project. The Explorer pane on the left lists several Java files, including LongProg6.java. The main editor displays the code for LongProg6.java, which defines a class LongProg6 with a main method. Inside the main method, a long variable 'a' is assigned the value 1255764, and a float variable 'b' is declared. A comment indicates implicit type casting from long to float. The variable 'b' is assigned the value of 'a', and the result is printed to the console. The Output pane at the bottom shows the execution of the program, displaying the value 1255764.0.

```
class LongProg6 {  
    public static void main(String[] args) {  
        long a = 1255764;  
        float b;  
        //implicit type casting  
        b = a;  
        System.out.println(b);  
    }  
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

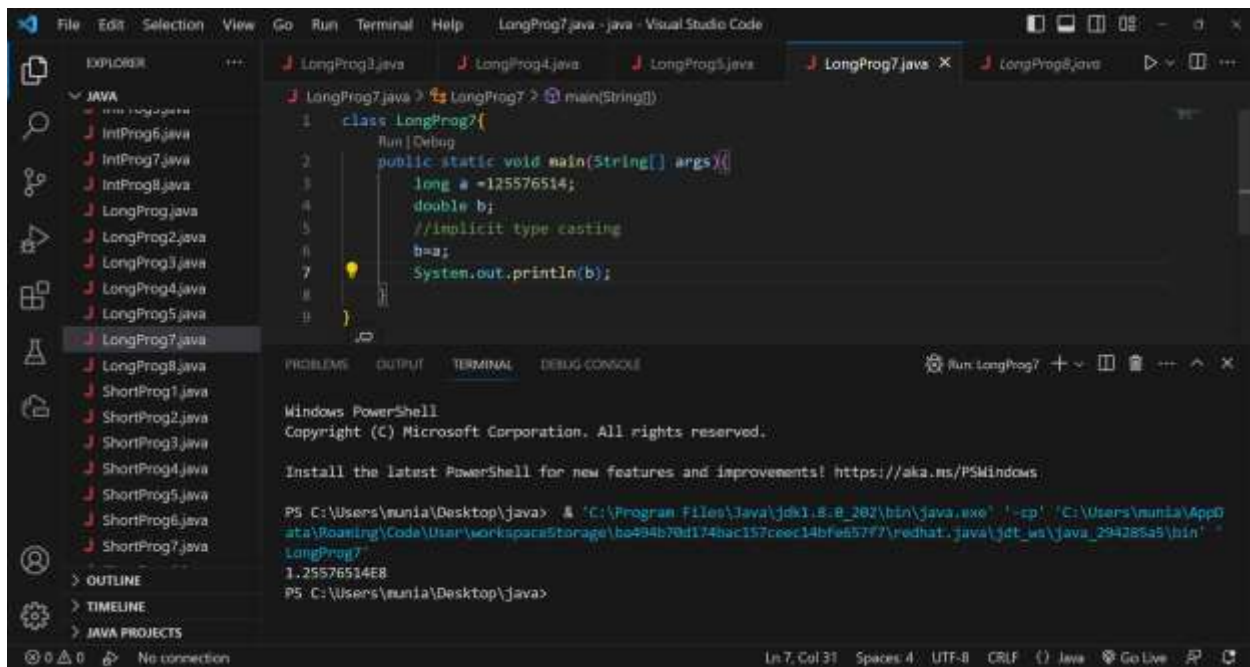
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\1ba494b70d174bac157ceec14bfe657f7\redhat.java\jdt_ws\java_294285a5\bin" "LongProg6"
1255764.0
PS C:\Users\munia\Desktop\java>

Program 39: (long to double)

```
class LongProg7 {  
    public static void main(String[] args) {  
        long a = 1255765141;  
        double b;  
        //implicit type casting  
        b = a;  
        System.out.println(b);  
    }  
}
```


Implicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, with LongProg7.java selected. The Editor pane shows the code for LongProg7.java, which defines a class LongProg7 with a main method. The code is as follows:

```
1 class LongProg7 {  
2     public static void main(String[] args) {  
3         long a = 125576514L;  
4         double b;  
5         //implicit type casting  
6         b = a;  
7         System.out.println(b);  
8     }  
9 }
```

The Output pane at the bottom shows the result of running the program:

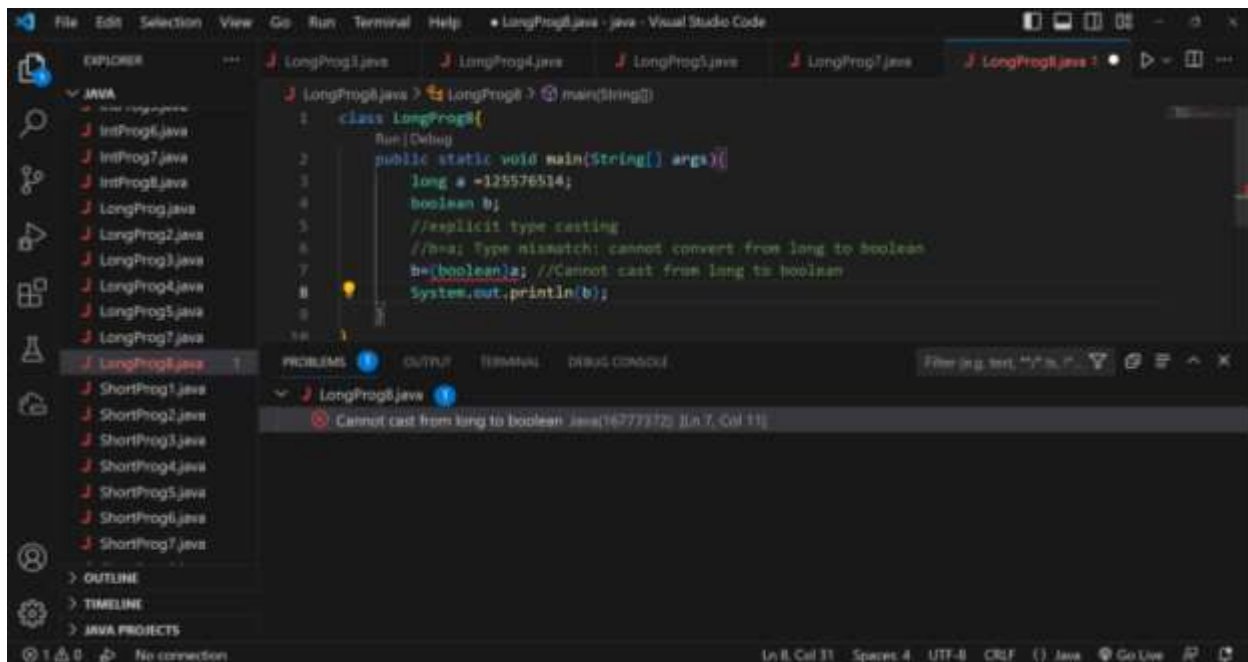
```
Run: LongProg7  
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Local\Roaming\Code\User\workspaceStorage\b0494b79d174bac157ceec14bfe657f7\redhat.java\jdk_ws\java_294285a5\bin" "LongProg7"  
1.25576514E8  
PS C:\Users\munia\Desktop\java>
```

Program 40: (long to boolean)

```
class LongProg8 {  
    public static void main(String[] args) {  
        long a = 125576514L;  
        boolean b;  
        //explicit type casting  
        //b=a; Type mismatch: cannot convert from long to boolean  
        b=(boolean)a; //Cannot cast from long to boolean  
        System.out.println(b);  
    }  
}
```

Casting cannot possible

Output:

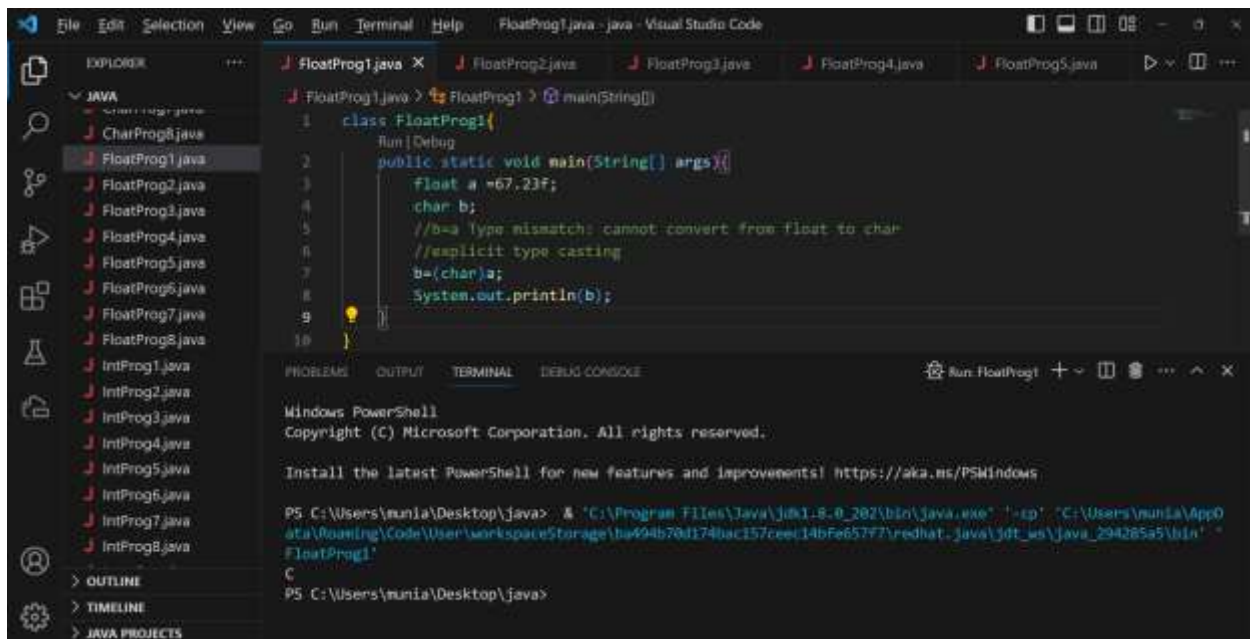


Program 41: (float to char)

```
class FloatProg1 {  
    public static void main(String[] args) {  
        float a = 67.23f;  
        char b;  
        //b=a Type mismatch: cannot convert from float to char  
        //explicit type casting  
        b=(char)a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including FloatProg1.java through FloatProg8.java. The main editor displays FloatProg1.java with the following code:

```
1 class FloatProg1 {
2     public static void main(String[] args) {
3         float a = 67.23f;
4         char b;
5         //b=a Type mismatch: cannot convert from float to char
6         //explicit type casting
7         b=(char)a;
8         System.out.println(b);
9     }
10 }
```

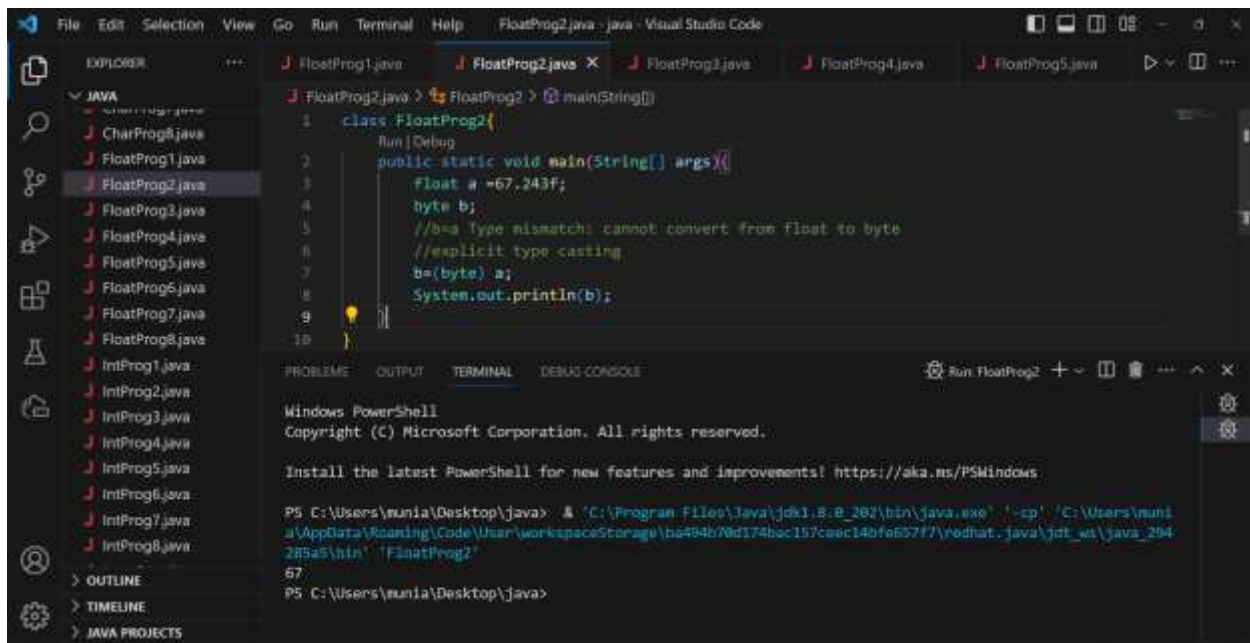
The error message is highlighted in red. The bottom panel shows the TERMINAL output, which includes the command to run the program and the output of the Java compiler.

Program 42: (float to byte)

```
class FloatProg2 {
    public static void main(String[] args) {
        float a = 67.243f;
        byte b;
        //b=a Type mismatch: cannot convert from float to byte
        //explicit type casting
        b=(byte) a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including FloatProg1.java through FloatProg8.java. The main editor displays FloatProg2.java with the following code:

```
1 class FloatProg2 {  
2     public static void main(String[] args) {  
3         float a = 67.243f;  
4         byte b;  
5         //b=a Type mismatch: cannot convert from float to byte  
6         //explicit type casting  
7         b = (byte) a;  
8         System.out.println(b);  
9     }  
10 }
```

The bottom panel shows the TERMINAL output, which displays the command to run the program and the resulting output:

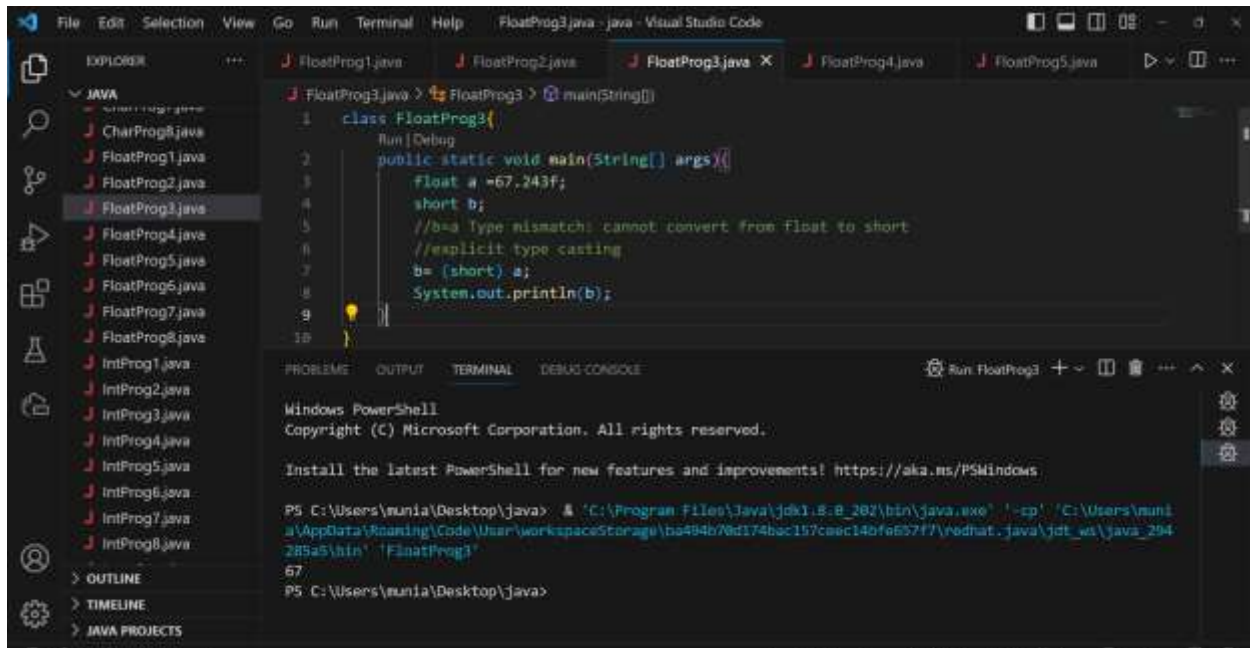
```
PS C:\Users\munia\Desktop\java> java -cp 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c9ec140fe657f7\redhat.java\jdk_wa\java_294285a5\bin' 'FloatProg2'  
67  
PS C:\Users\munia\Desktop\java>
```

Program 43: (float to short)

```
class FloatProg3 {  
    public static void main(String[] args) {  
        float a = 67.243f;  
        short b;  
        //b=a Type mismatch: cannot convert from float to short  
        //explicit type casting  
        b = (short) a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, with 'FloatProg3.java' selected. The main editor displays the code for 'FloatProg3.java':

```
1 class FloatProg3{
2     public static void main(String[] args){
3         float a =67.243f;
4         short b;
5         //b=a Type mismatch: cannot convert from float to short
6         //explicit type casting
7         b= (short) a;
8         System.out.println(b);
9     }
10 }
```

The bottom panel shows the 'TERMINAL' tab with the following output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

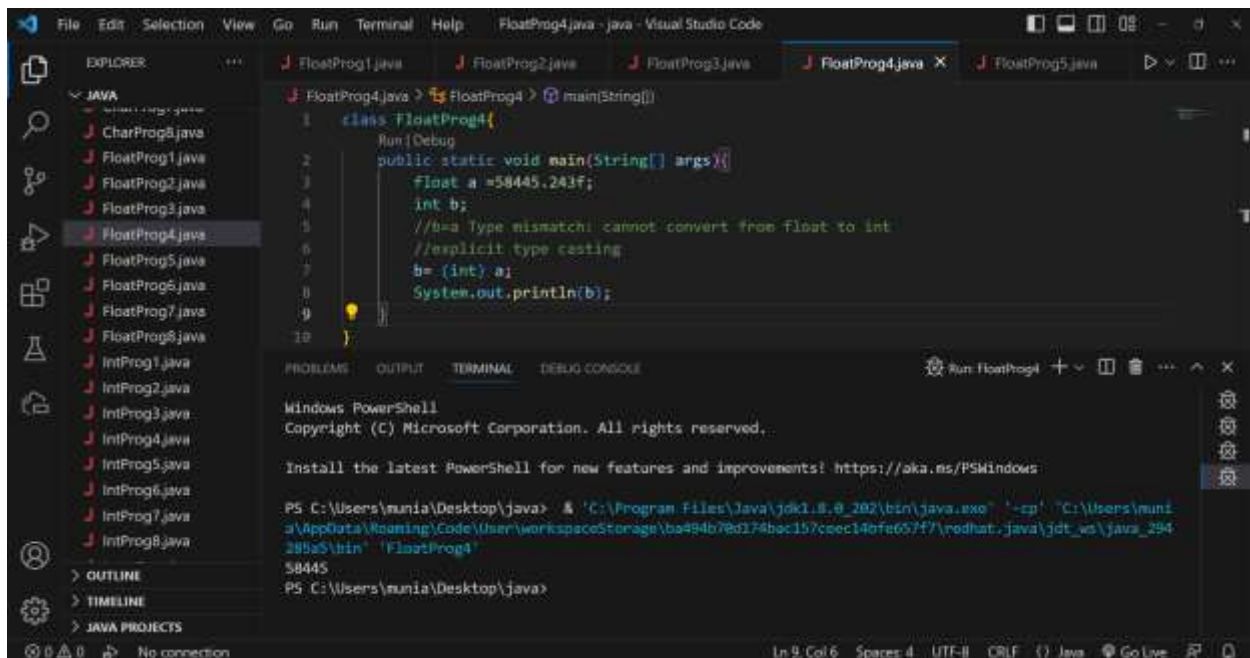
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c9ec140fe657f7\redhat.java\jdt_ws\java_294285a5\bin" "FloatProg3"
67
PS C:\Users\munia\Desktop\java>
```

Program 44: (float to int)

```
class FloatProg4{
    public static void main(String[] args){
        float a =58445.243f;
        int b;
        //b=a Type mismatch: cannot convert from float to int
        //explicit type casting
        b= (int) a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, with 'FloatProg4.java' selected. The main editor displays the code for 'FloatProg4.java':

```
1 class FloatProg4 {
2     public static void main(String[] args) {
3         float a = 58445.243f;
4         int b;
5         //b=a Type mismatch: cannot convert from float to int
6         //explicit type casting
7         b = (int) a;
8         System.out.println(b);
9     }
10 }
```

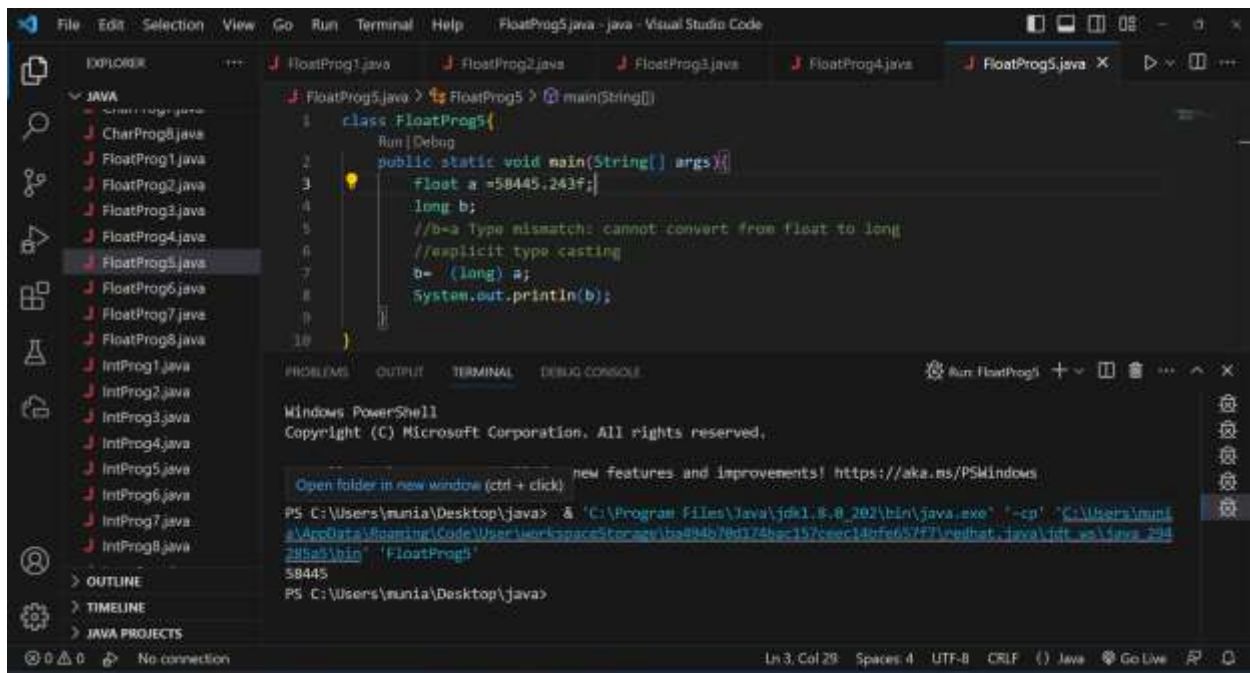
The output window at the bottom shows the command prompt running the program, resulting in the output '58445'.

Program 45: (float to long)

```
class FloatProg5 {
    public static void main(String[] args) {
        float a = 58445.243f;
        long b;
        //b=a Type mismatch: cannot convert from float to long
        //explicit type casting
        b = (long) a;
        System.out.println(b);
    }
}
```


Explicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named `FloatProg5.java`. The code defines a class `FloatProg5` with a `main` method. Inside the `main` method, a `float` variable `a` is initialized with the value `58445.243f`. A `long` variable `b` is declared, and a comment indicates a type mismatch: `//b=a Type mismatch: cannot convert from float to long`. To resolve this, explicit type casting is used: `b= (long) a;`. Finally, the value of `b` is printed using `System.out.println(b);`. The terminal at the bottom shows the command to run the program and the output `58445`.

```
class FloatProg5 {  
    public static void main(String[] args) {  
        float a = 58445.243f;  
        long b;  
        //b=a Type mismatch: cannot convert from float to long  
        //explicit type casting  
        b= (long) a;  
        System.out.println(b);  
    }  
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

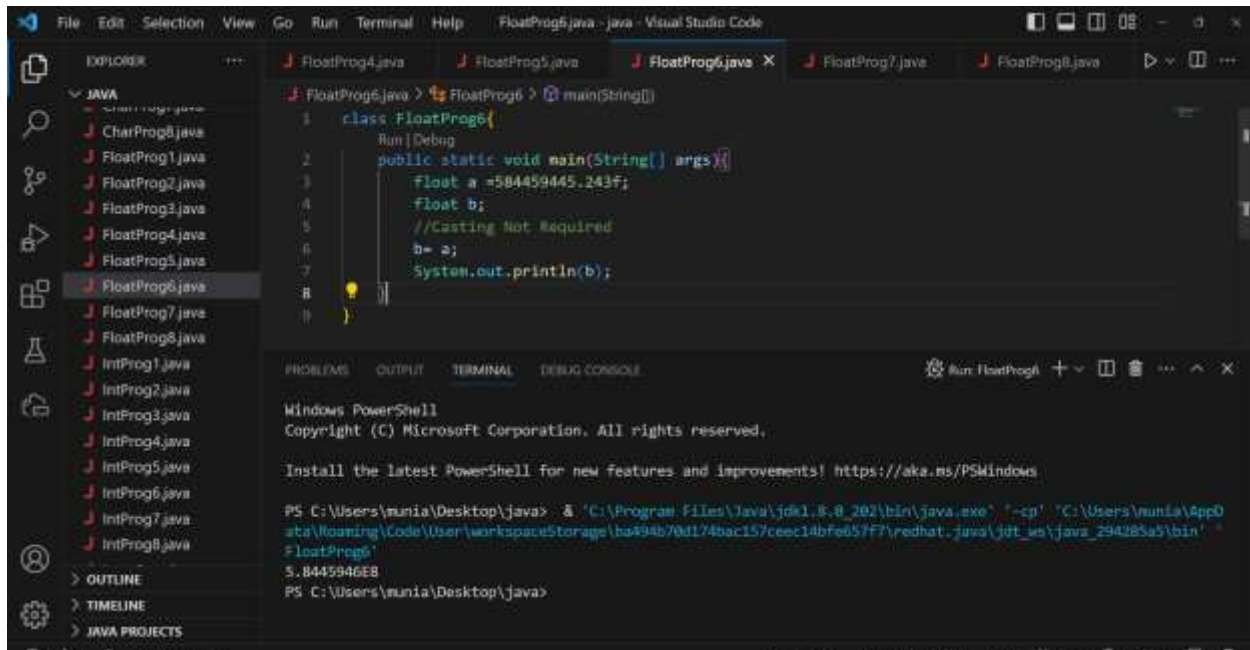
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Local\Temp\Code\User\workspaceStorage\1ba894b78d174ba31574ee14bfe657f71\redhat.java\jdt_ws\java_24d285a5\bin' 'FloatProg5'
58445
PS C:\Users\munia\Desktop\java>

Program 46: (float to float)

```
class FloatProg6 {  
    public static void main(String[] args) {  
        float a = 584459445.243f;  
        float b;  
        //Casting Not Required  
        b = a;  
        System.out.println(b);  
    }  
}
```


Casting not required

Output:



```
File Edit Selection View Go Run Terminal Help FloatProg6.java - java - Visual Studio Code
EXPLORER
  JAVA
    FloatProg1.java
    FloatProg2.java
    FloatProg3.java
    FloatProg4.java
    FloatProg5.java
    FloatProg6.java
    FloatProg7.java
    FloatProg8.java
    IntProg1.java
    IntProg2.java
    IntProg3.java
    IntProg4.java
    IntProg5.java
    IntProg6.java
    IntProg7.java
    IntProg8.java
  OUTPUT
  TIMELINE
  JAVA PROJECTS

FloatProg6.java > FloatProg6 > main(String[])
1 class FloatProg6{
2     public static void main(String[] args){
3         float a =584459445.243f;
4         float b;
5         //Casting Not Required
6         b= a;
7         System.out.println(b);
8     }
9 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Run FloatProg6
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_202\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\b4694b7bd174bac157ceec14bfe657f7\redhat.java\jdk_ws\java_294285a5\bin" "FloatProg6"
5.8445946E8
PS C:\Users\munia\Desktop\java>
```

Program 47: (float to double)

```
class FloatProg7{

    public static void main(String[] args){

        float a =585424459445.243f;

        double b;

        //implicit type casting

        b= a;

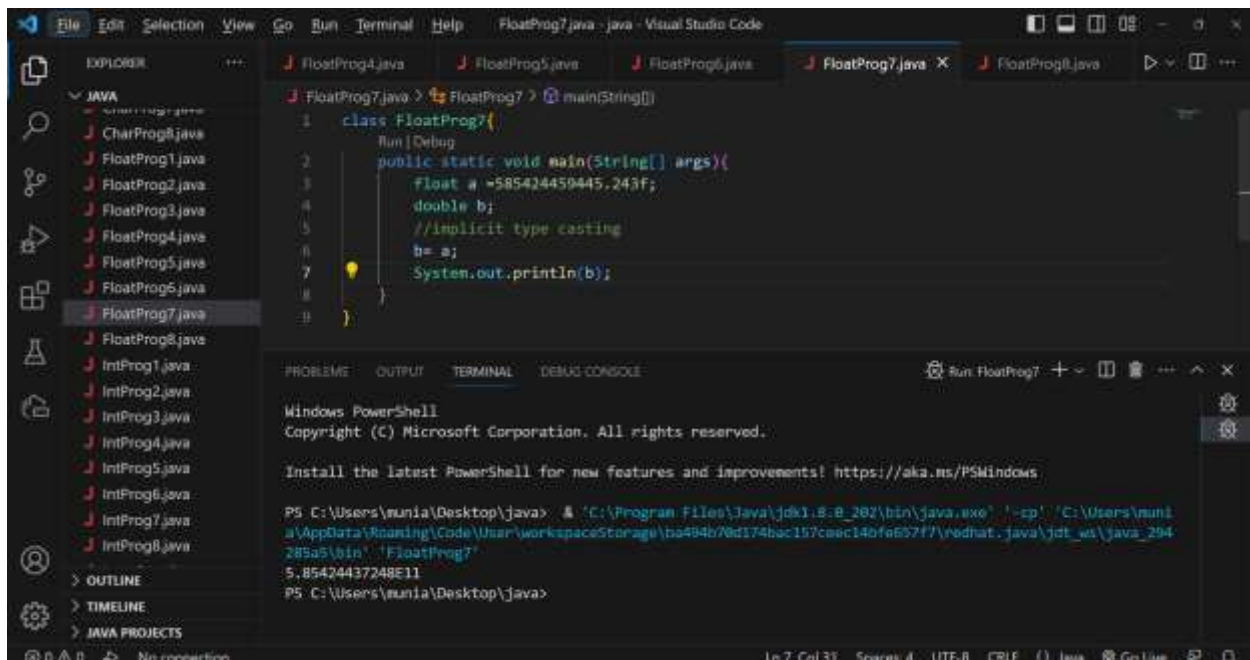
        System.out.println(b);

    }

}
```

Implicit type casting

Output:



```
File Edit Selection View Go Run Terminal Help FloatProg7.java - java - Visual Studio Code
EXPLORER
  JAVA
    FloatProg1.java
    FloatProg2.java
    FloatProg3.java
    FloatProg4.java
    FloatProg5.java
    FloatProg6.java
    FloatProg7.java
    FloatProg8.java
    IntProg1.java
    IntProg2.java
    IntProg3.java
    IntProg4.java
    IntProg5.java
    IntProg6.java
    IntProg7.java
    IntProg8.java
  OUTLINE
  TIMELINE
  JAVA PROJECTS

FloatProg7.java > FloatProg7 > main(String[])
1 class FloatProg7{
2     public static void main(String[] args){
3         float a = 585424459445.243f;
4         double b;
5         //implicit type casting
6         b = a;
7         System.out.println(b);
8     }
9 }

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Run: FloatProg7
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

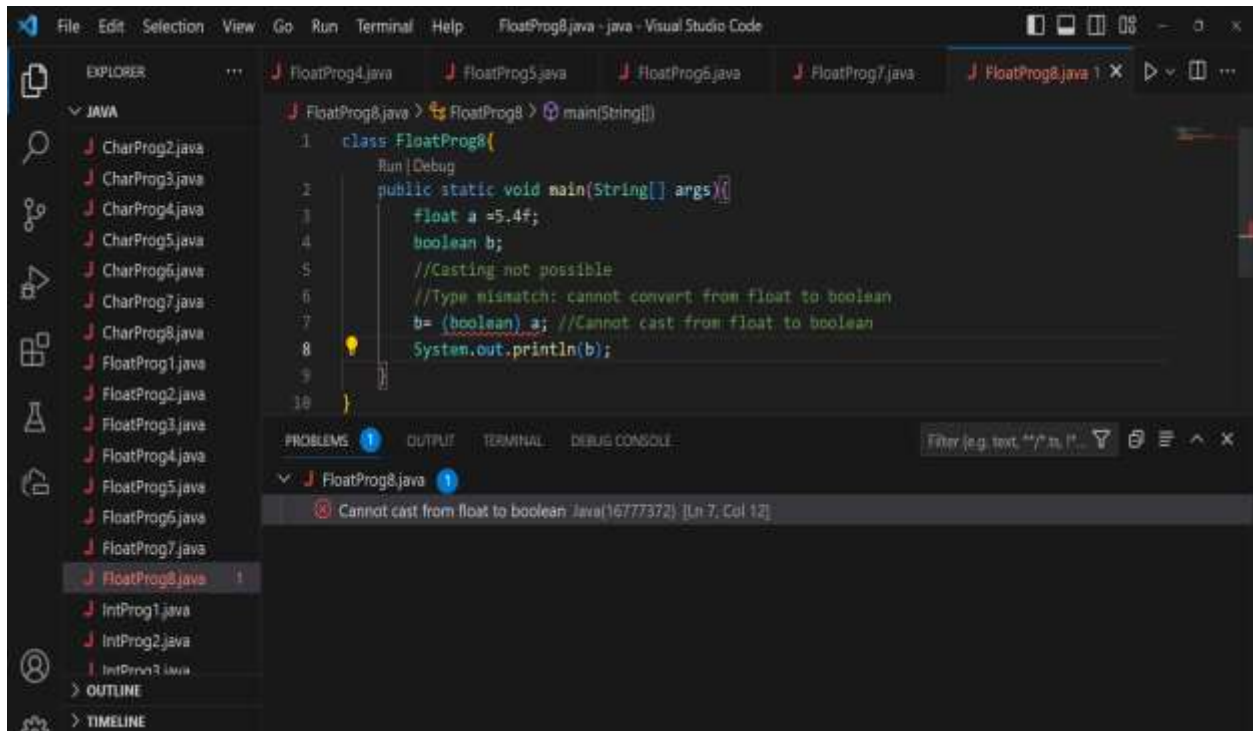
PS C:\Users\munia\Desktop\java> & "C:\Program Files\Java\jdk1.8.0_282\bin\java.exe" "-cp" "C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba494b70d174bac157c9ec140fe857f7\redhat.java\jdk_wa\java_294285a5\bin" "FloatProg7"
5.85424437248E11
PS C:\Users\munia\Desktop\java>
```

Program 48: (float to Boolean)

```
class FloatProg8 {
    public static void main(String[] args) {
        float a = 5.4f;
        boolean b;
        //Casting not possible
        //Type mismatch: cannot convert from float to boolean
        b = (boolean) a; //Cannot cast from float to boolean
        System.out.println(b);
    }
}
```

Casting not possible

Output:

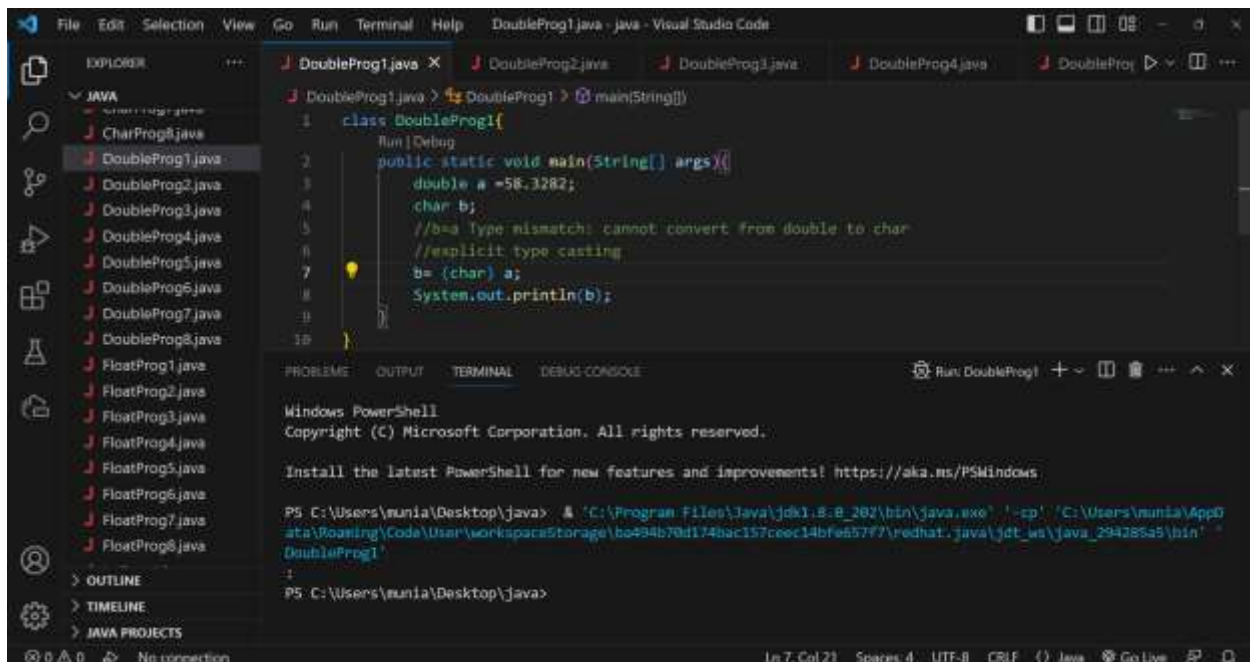


Program 49: (double to char)

```
class DoubleProg1 {  
    public static void main(String[] args){  
        double a = 58.3282;  
        char b;  
        //b=a Type mismatch: cannot convert from double to char  
        //explicit type casting  
        b = (char) a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named DoubleProg1.java. The code defines a class DoubleProg1 with a main method. Inside the main method, a double variable 'a' is assigned the value 56.3282. A char variable 'b' is declared, and a comment indicates a type mismatch: '//b=a Type mismatch: cannot convert from double to char'. The code then uses explicit type casting: 'b = (char) a;'. Finally, 'System.out.println(b);' is called. The terminal at the bottom shows the command to run the program: 'PS C:\Users\munia\Desktop\java> java -cp 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' 'C:\Users\munia\AppData\Local\Roaming\Coda\User\workspaceStorage\ba94b79d174bac157ceec14bfe657f7\redhat.java\jdk_ws\java_294285a5\bin' DoubleProg1'. The output in the terminal is '56'.

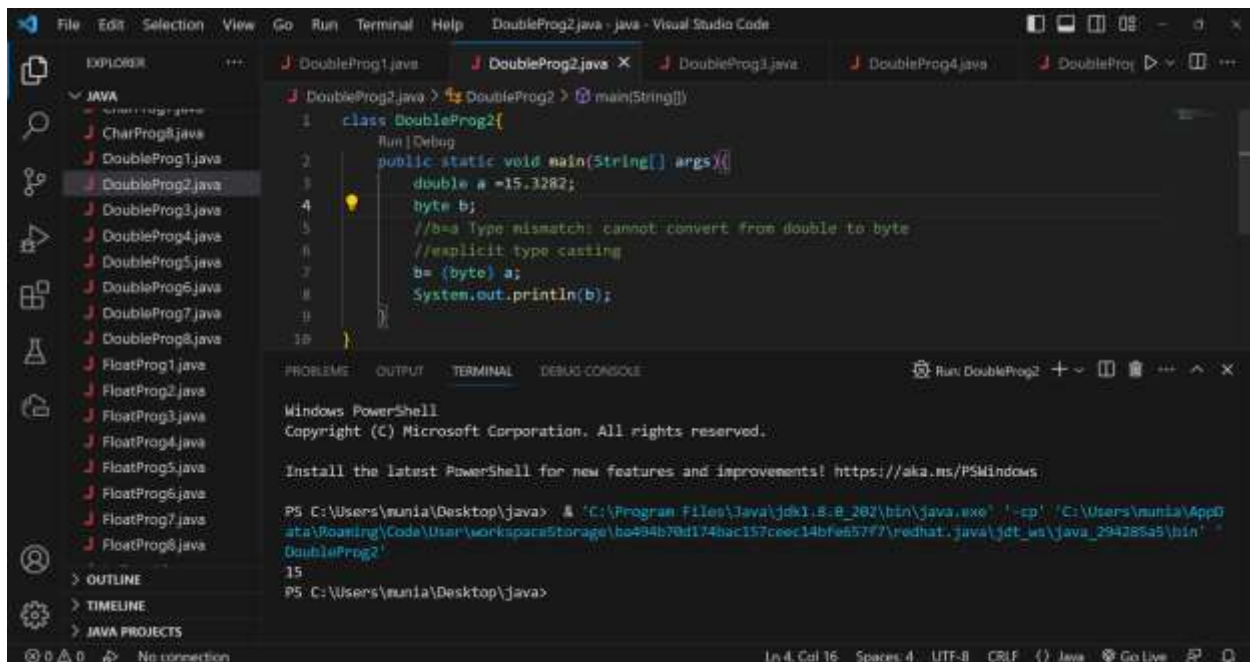
```
class DoubleProg1 {
    public static void main(String[] args) {
        double a = 56.3282;
        char b;
        //b=a Type mismatch: cannot convert from double to char
        //explicit type casting
        b = (char) a;
        System.out.println(b);
    }
}
```

Program 50: (double to byte)

```
class DoubleProg2 {
    public static void main(String[] args) {
        double a = 15.3282;
        byte b;
        //b=a Type mismatch: cannot convert from double to byte
        //explicit type casting
        b = (byte) a;
        System.out.println(b);
    }
}
```

Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including DoubleProg1.java through DoubleProg8.java. The main editor displays the code for DoubleProg2.java, which contains the following code:

```
1 class DoubleProg2 {  
2     public static void main(String[] args) {  
3         double a = 15.3282;  
4         byte b;  
5         //b=a Type mismatch: cannot convert from double to byte  
6         //explicit type casting  
7         b = (byte) a;  
8         System.out.println(b);  
9     }  
10 }
```

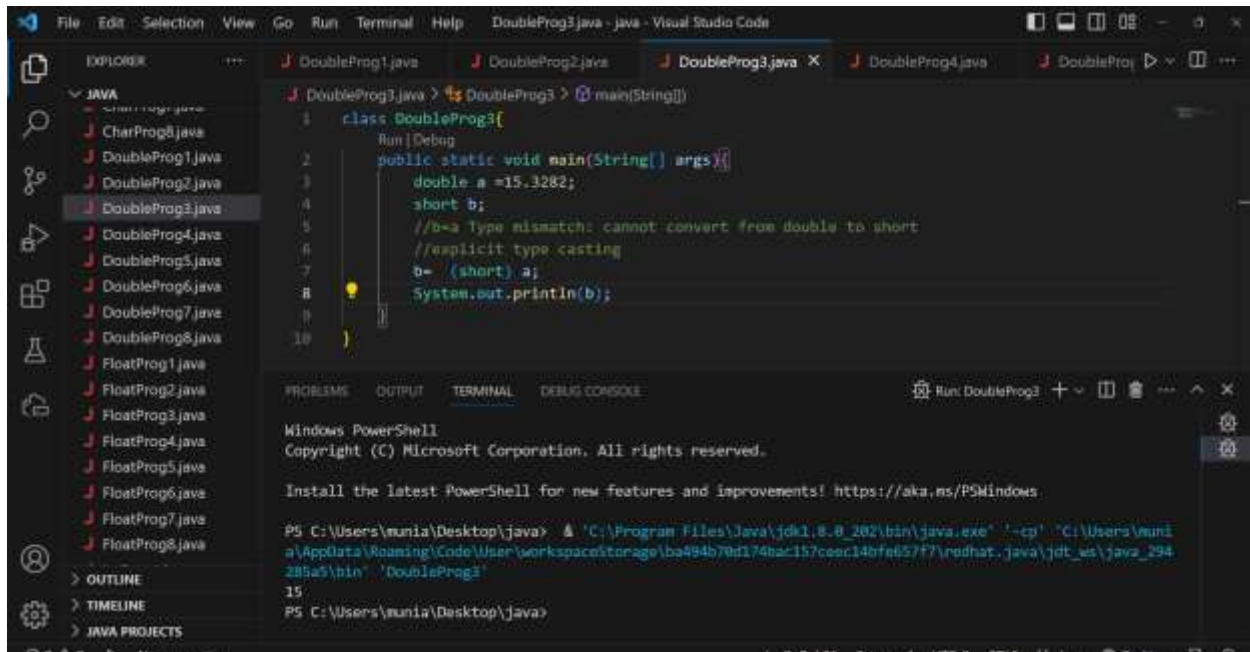
The code is annotated with a yellow lightbulb icon on line 4, indicating a warning. The output pane at the bottom shows the command prompt output for running DoubleProg2, which displays the value 15.

Program 51: (double to short)

```
class DoubleProg3 {  
    public static void main(String[] args) {  
        double a = 15.3282;  
        short b;  
        //b=a Type mismatch: cannot convert from double to short  
        //explicit type casting  
        b = (short) a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:

A screenshot of the Visual Studio Code editor. The Explorer sidebar on the left shows a project named 'DoubleProg' with files DoubleProg1.java through DoubleProg8.java. The main editor window displays DoubleProg3.java with the following code:

```
1 class DoubleProg3{
2     public static void main(String[] args){
3         double a =15.3282;
4         short b;
5         //b=a Type mismatch: cannot convert from double to short
6         //explicit type casting
7         b= (short) a;
8         System.out.println(b);
9     }
10 }
```

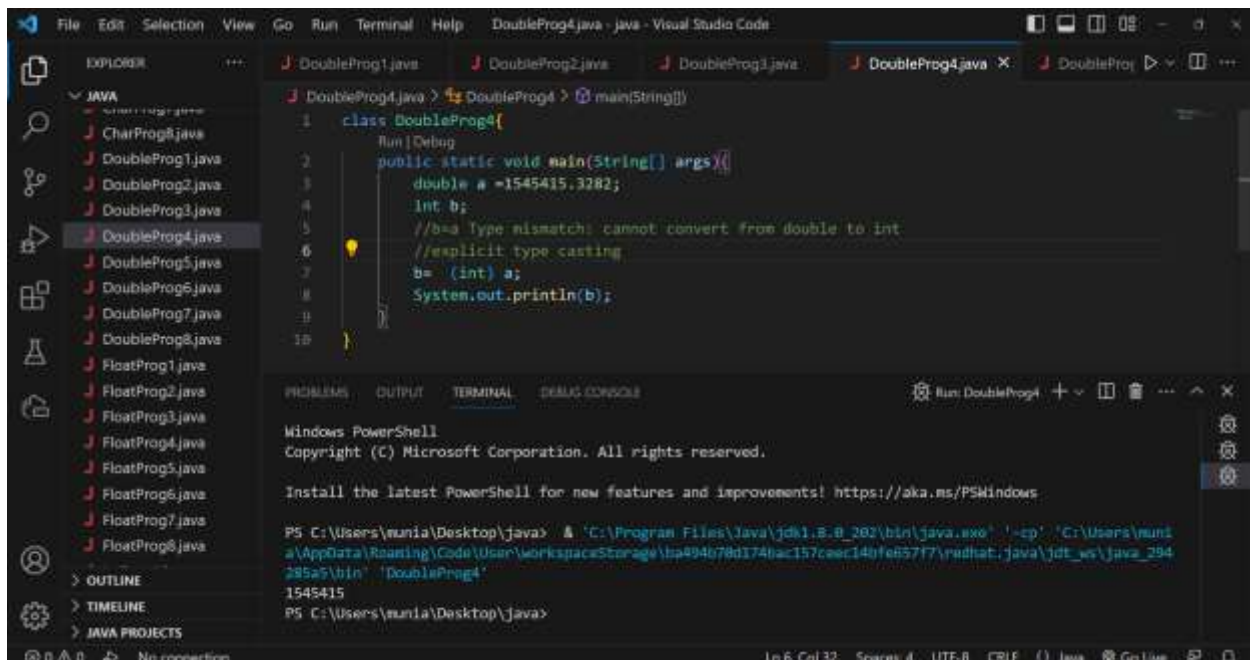
The bottom panel shows the 'TERMINAL' tab with a Windows PowerShell prompt. The command executed is: `PS C:\Users\munia\Desktop\java> java -cp 'C:\Program Files\Java\jdk1.8.0_202\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspacestorage\ba494b70d174bac157ceac1407e657f7\redhat_java\jdt_ws\java_294285a5\bin' 'DoubleProg3'`. The output is: `15`.

Program 52: (double to int)

```
class DoubleProg4{
    public static void main(String[] args){
        double a =1545415.3282;
        int b;
        //b=a Type mismatch: cannot convert from double to int
        //explicit type casting
        b= (int) a;
        System.out.println(b);
    }
}
```


Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface with a Java file named DoubleProg4.java. The code defines a class DoubleProge with a main method. Inside the main method, a double variable 'a' is assigned the value 1545415.3282. An int variable 'b' is declared, and a comment indicates a type mismatch: '//b=a Type mismatch: cannot convert from double to int'. The code then uses explicit type casting: 'b= (int) a;'. Finally, 'System.out.println(b);' is called. The output in the terminal shows the value 1545415.

```
class DoubleProge{
    public static void main(String[] args){
        double a =1545415.3282;
        int b;
        //b=a Type mismatch: cannot convert from double to int
        //explicit type casting
        b= (int) a;
        System.out.println(b);
    }
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\munia\Desktop\java> A 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\1b494b78d174bac157c6ac14bfe857f7\radhat.java\jdt_ws\java_294285a5\bin' 'DoubleProg4'

1545415

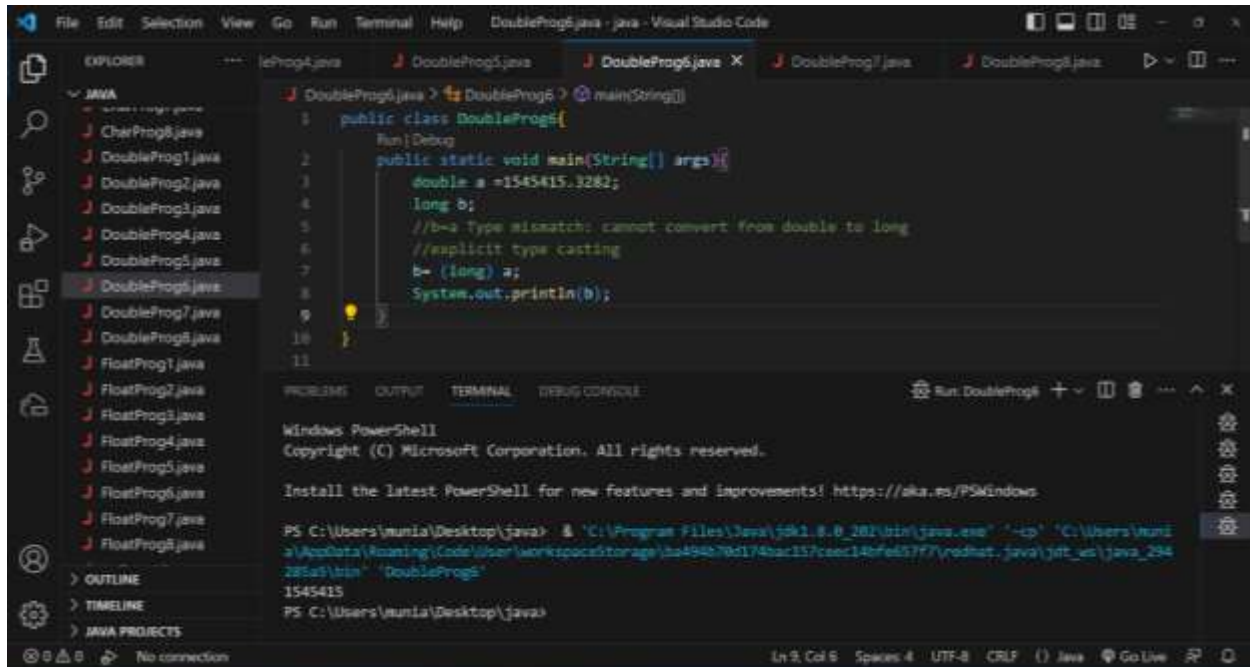
PS C:\Users\munia\Desktop\java>

Program 53: (double to long)

```
public class DoubleProg6{
    public static void main(String[] args){
        double a =1545415.3282;
        long b;
        //b=a Type mismatch: cannot work convert from double to long
        //explicit type casting
        b= (long) a;
        System.out.println(b);
    }
}
```


Explicit type casting

Output:



The screenshot shows the Visual Studio Code interface with a Java file named DoubleProg6.java. The code defines a public class DoubleProg6 with a main method. Inside the main method, a double variable 'a' is assigned the value 1545415.3282. A long variable 'b' is declared, and a comment indicates a type mismatch: '//b=a Type mismatch: cannot convert from double to long'. The code then performs explicit type casting: 'b= (long) a;'. Finally, 'System.out.println(b);' is called. The output in the terminal shows the value 1545415. The Explorer sidebar on the left lists several other Java files, including CharProg8.java, DoubleProg1.java through DoubleProg8.java, FloatProg1.java through FloatProg8.java, and OUTLINE, TIMELINE, and JAVA PROJECTS. The bottom status bar indicates 'Ln 9, Col 6, Spaces: 4, UTF-8, CRLF, Java, Go Live'.

```
1 public class DoubleProg6 {
2     public static void main(String[] args) {
3         double a = 1545415.3282;
4         long b;
5         //b=a Type mismatch: cannot convert from double to long
6         //explicit type casting
7         b= (long) a;
8         System.out.println(b);
9     }
10 }
11
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

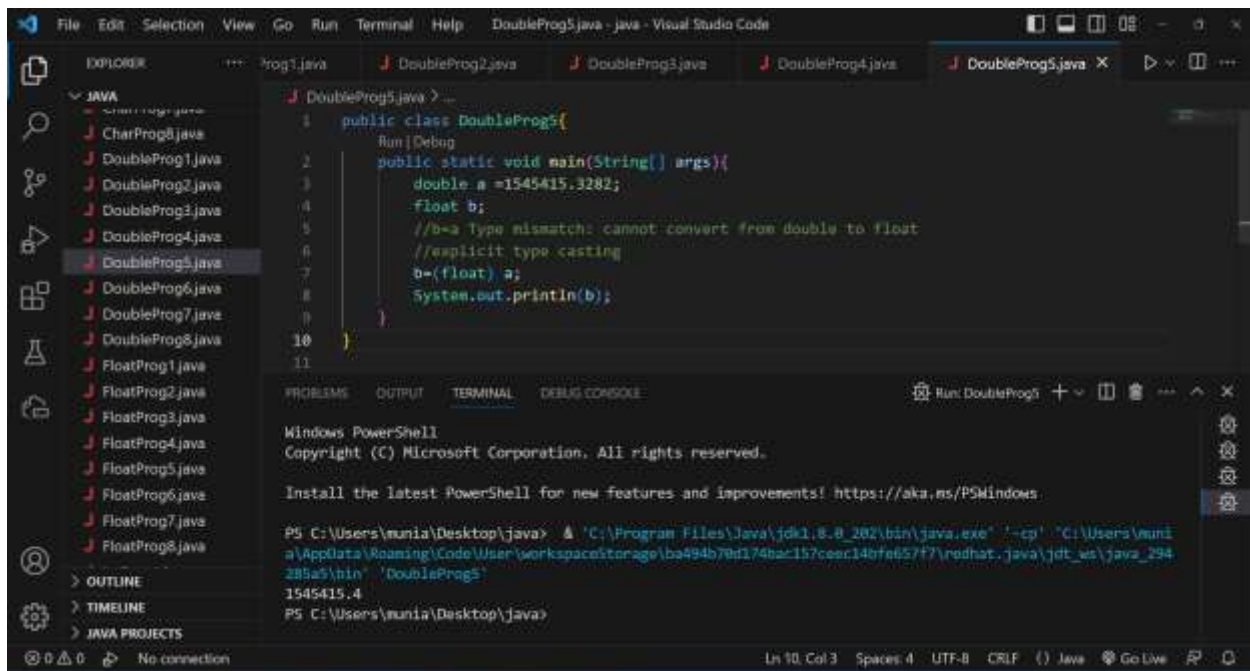
PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_281\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspaceStorage\ba49487bd174ba137c0c14bf6d57f7\rodhat_java\jrt_ws\java_294285a5\bin' 'DoubleProg6'
1545415
PS C:\Users\munia\Desktop\java>

Program 54: (double to float)

```
public class DoubleProg5 {  
    public static void main(String[] args) {  
        double a = 1545415.3282;  
        float b;  
        //b=a Type mismatch: cannot convert from double to float  
        //explicit type casting  
        b=(float) a;  
        System.out.println(b);  
    }  
}
```

Explicit type casting

Output:



```
1 public class DoubleProg5{
2     public static void main(String[] args){
3         double a =1545415.3282;
4         float b;
5         //b=a Type mismatch: cannot convert from double to float
6         //explicit type casting
7         b=(float) a;
8         System.out.println(b);
9     }
10 }
11
```

Run: DoubleProg5

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

PS C:\Users\munia\Desktop\java> & 'C:\Program Files\Java\jdk1.8.0_282\bin\java.exe' '-cp' 'C:\Users\munia\AppData\Roaming\Code\User\workspacestorage\ba494b70d174bac157ceac140fe657f7\redhat_java\jdt_ws\java_294285a5\bin' 'DoubleProg5'

1545415.4

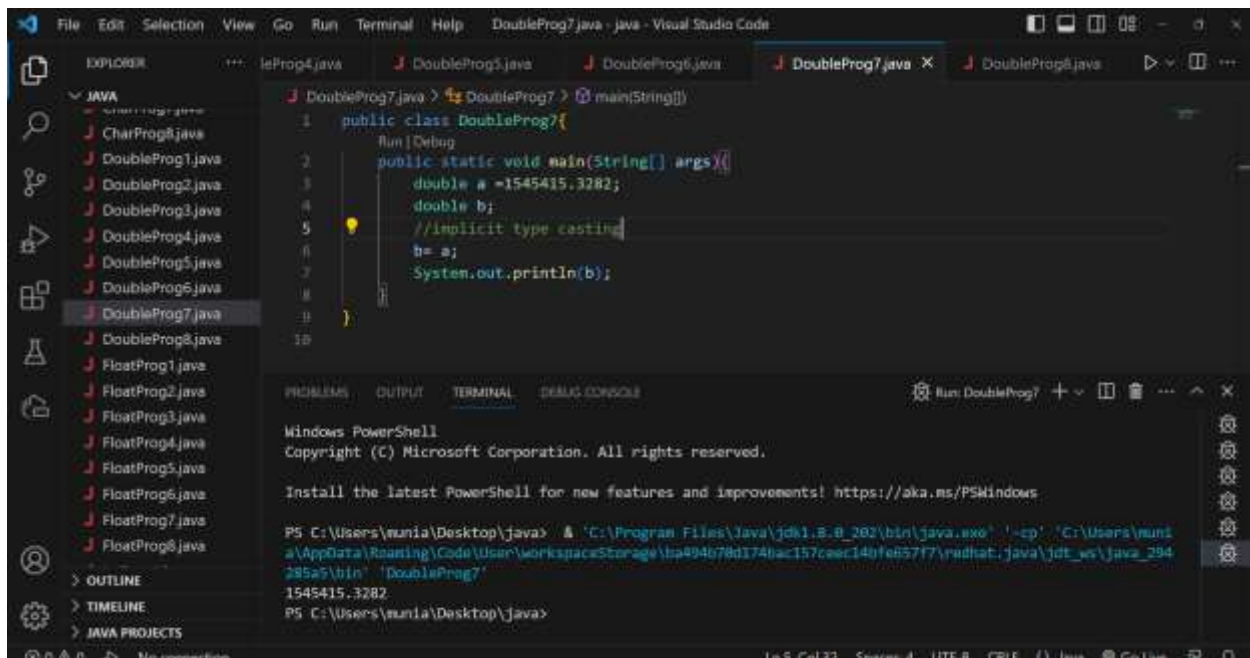
PS C:\Users\munia\Desktop\java>

Program 55: (double to double)

```
public class DoubleProg7{
    public static void main(String[] args){
        double a =1545415.3282;
        double b;
        //implicit type casting
        b = a;
        System.out.println(b);
    }
}
```

Implicit type casting

Output:



The screenshot shows the Visual Studio Code editor with a Java file named DoubleProg7.java. The code defines a class DoubleProg7 with a main method. Inside the main method, a double variable 'a' is assigned the value 1545415.3282. A boolean variable 'b' is then assigned the value of 'a', with a comment '//Implicit type casting'. Finally, 'b' is printed to the console. The terminal at the bottom shows the command to run the program and the output '1545415.3282'.

```
public class DoubleProg7 {  
    public static void main(String[] args) {  
        double a = 1545415.3282;  
        double b;  
        //Implicit type casting  
        b = a;  
        System.out.println(b);  
    }  
}
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! <https://aka.ms/PSWindows>

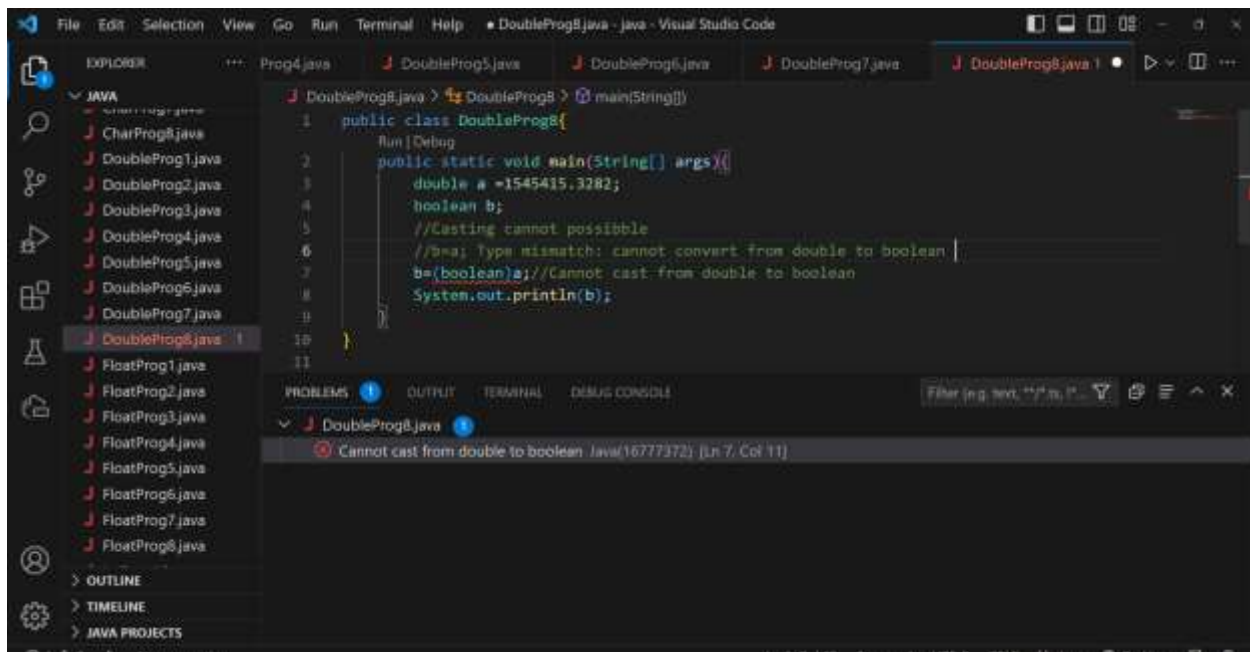
PS C:\Users\munia\Desktop\java> .\DoubleProg7.exe
1545415.3282
PS C:\Users\munia\Desktop\java>

Program 56: (double to boolean)

```
public class DoubleProg8 {  
    public static void main(String[] args) {  
        double a = 1545415.3282;  
        boolean b;  
        //Casting cannot possible  
        //b=a; Type mismatch: cannot convert from double to boolean  
        b=(boolean)a;//Cannot cast from double to boolean  
        System.out.println(b);  
    }  
}
```

Casting Cannot Possible

Output:

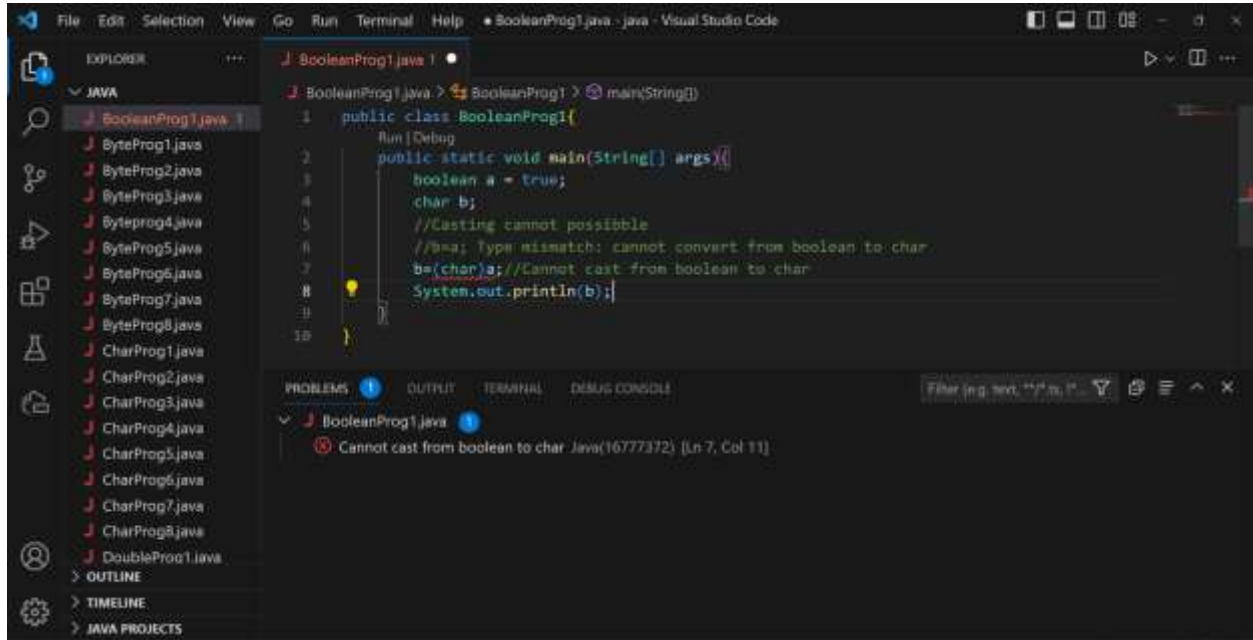


Program 57: (boolean to char)

```
public class BooleanProg1 {
    public static void main(String[] args) {
        boolean a = true;
        char b;
        //Casting cannot possible
        //b=a; Type mismatch: cannot convert from boolean to char
        b=(char)a; //Cannot cast from boolean to char
        System.out.println(b);
    }
}
```

Casting Cannot Possible

Output:

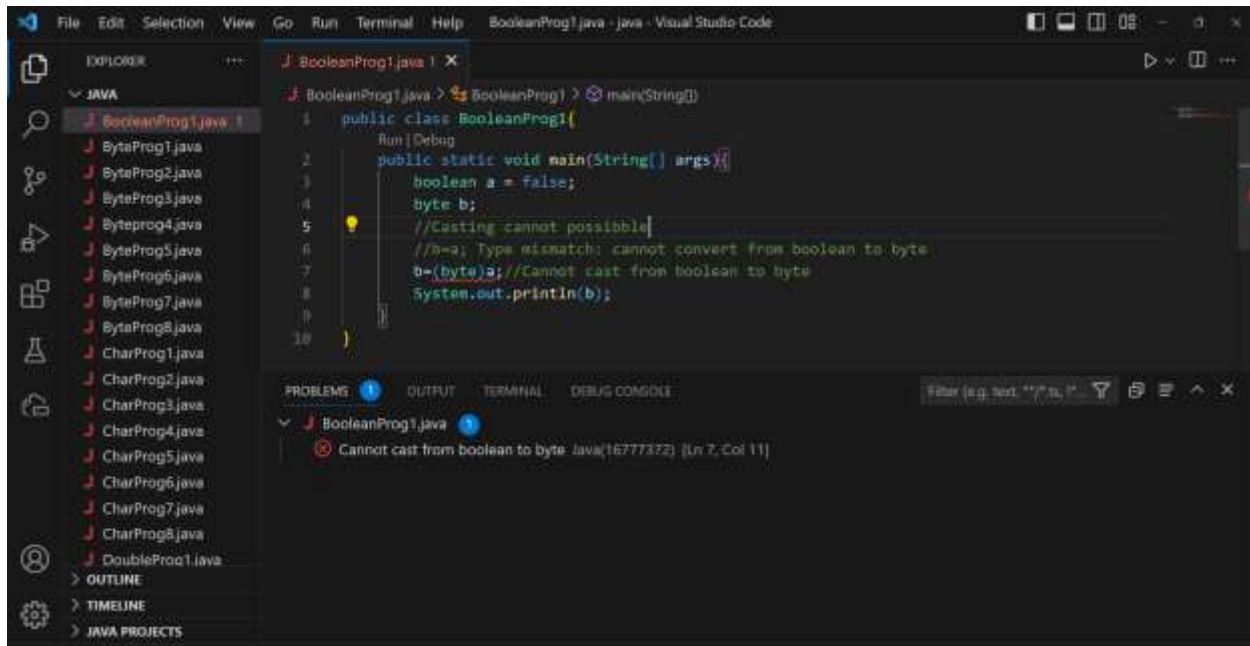


Program 58: (boolean to byte)

```
public class BooleanProg1 {  
    public static void main(String[] args){  
        boolean a = false;  
        byte b;  
        //Casting cannot possible  
        //b=a; Type mismatch: cannot convert from boolean to byte  
        b=(byte)a; //Cannot cast from boolean to byte  
        System.out.println(b);  
    }  
}
```

Casting Cannot Possible

Output:

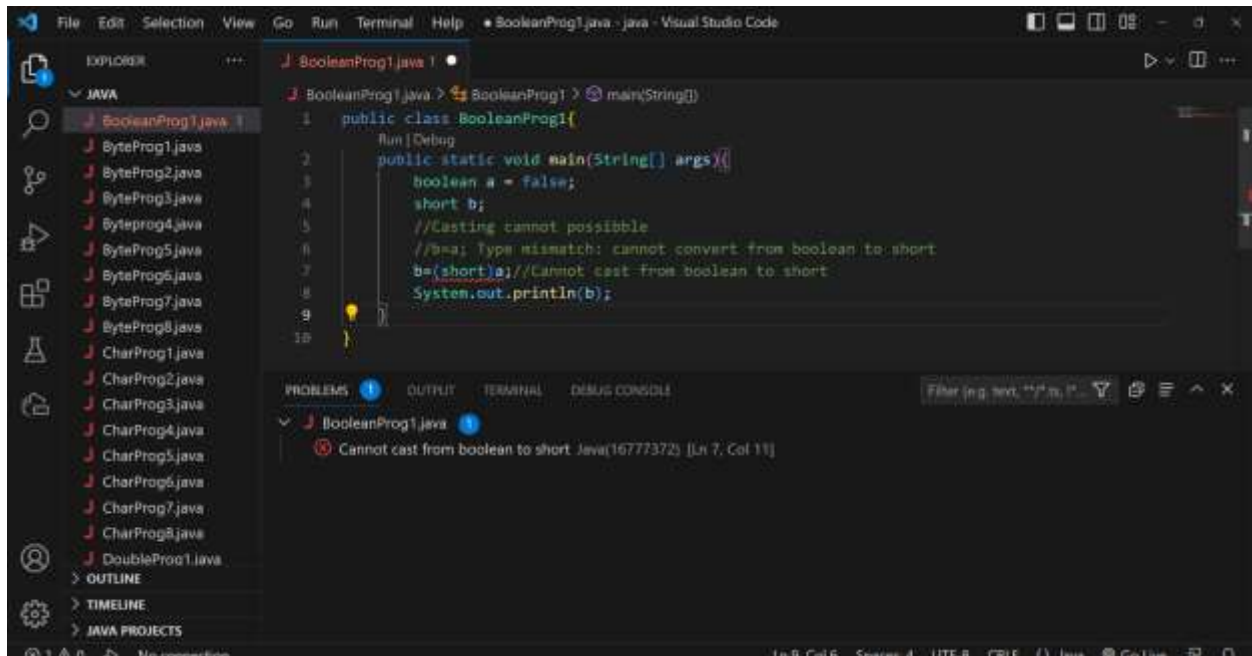


Program 59: (boolean to short)

```
public class BooleanProg1 {
    public static void main(String[] args) {
        boolean a = false;
        short b;
        //Casting cannot possible
        //b=a; Type mismatch: cannot convert from boolean to short
        b=(short)a; //Cannot cast from boolean to short
        System.out.println(b);
    }
}
```

Casting Cannot Possible

Output:

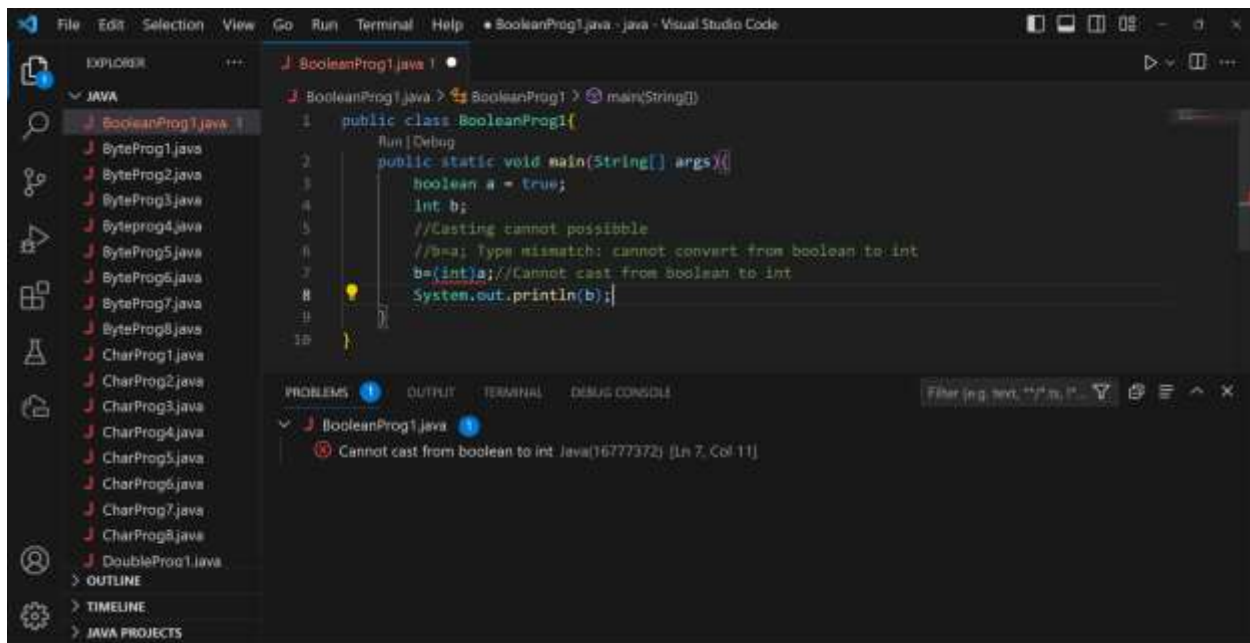


Program 60: (boolean to int)

```
public class BooleanProg1 {  
    public static void main(String[] args) {  
        boolean a = true;  
        int b;  
        //Casting cannot possible  
        //b=a; Type mismatch: cannot convert from boolean to int  
        b=(int)a; //Cannot cast from boolean to int  
        System.out.println(b);  
    }  
}
```


Casting Cannot Possible

Output:

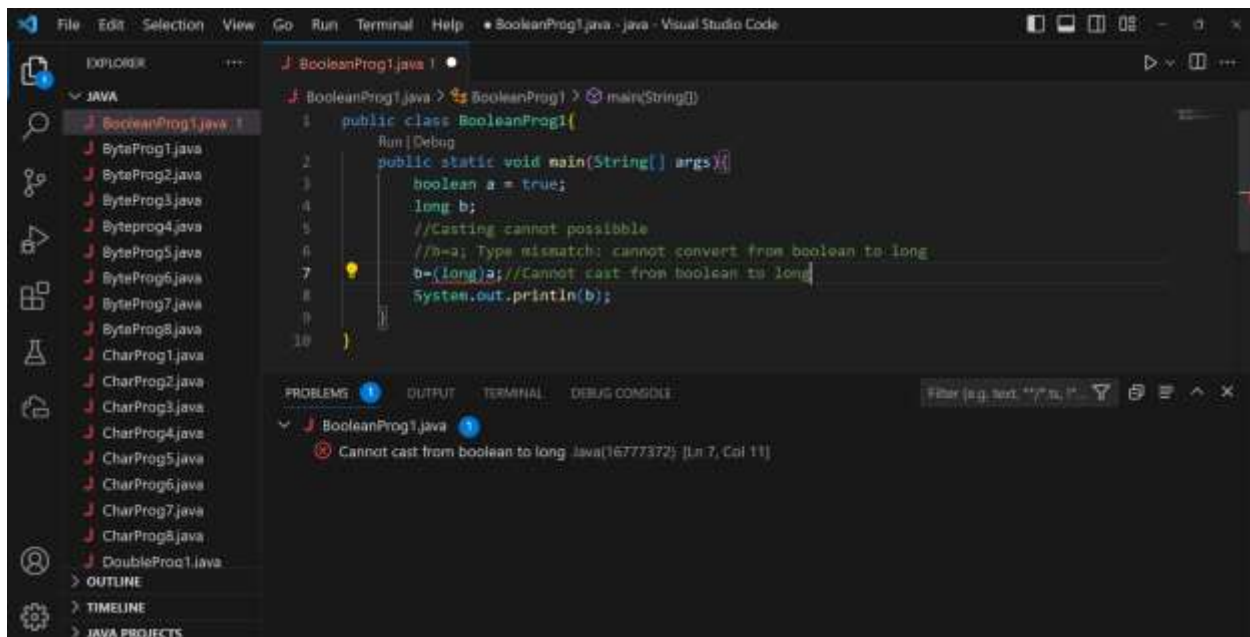


Program 61: (boolean to long)

```
public class BooleanProg1 {  
    public static void main(String[] args) {  
        boolean a = true;  
        long b;  
        //Casting cannot possible  
        //b=a; Type mismatch: cannot convert from boolean to long  
        b=(long)a; //Cannot cast from boolean to long  
        System.out.println(b);  
    }  
}
```

Casting Cannot Possible

Output:

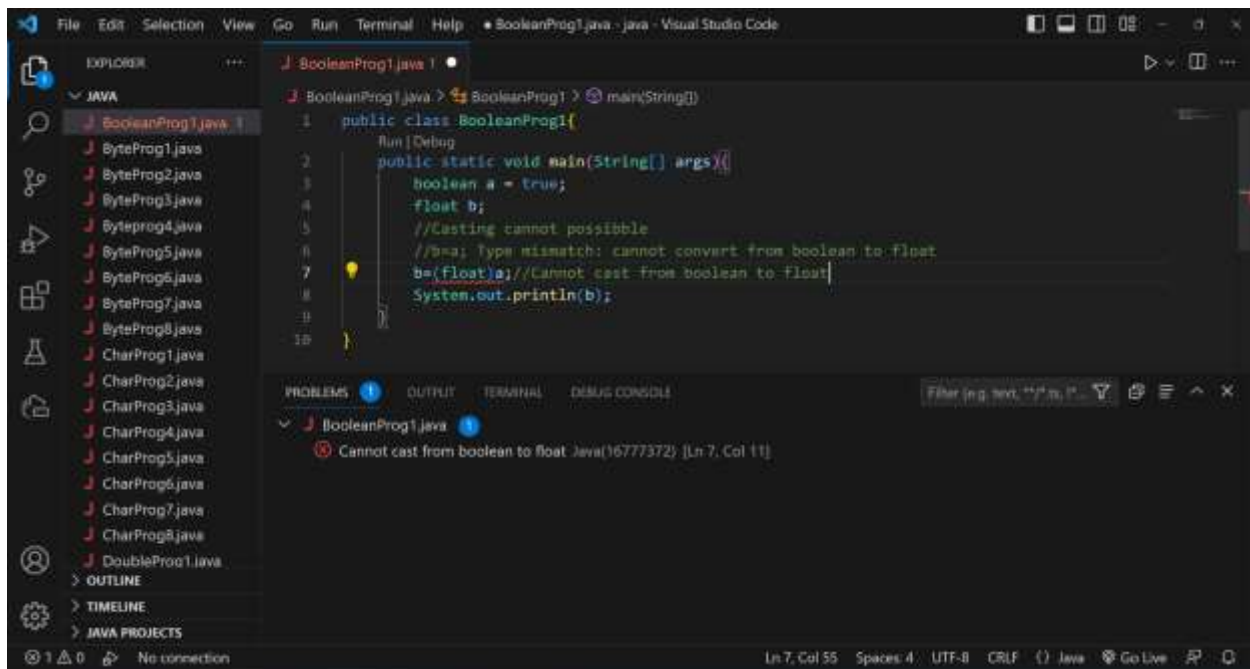


Program 62: (boolean to float)

```
public class BooleanProg1 {
    public static void main(String[] args){
        boolean a = true;
        float b;
        //Casting cannot possible
        //b=a; Type mismatch: cannot convert from boolean to float
        b=(float)a; //Cannot cast from boolean to float
        System.out.println(b);
    }
}
```

Casting Cannot Possible

Output:

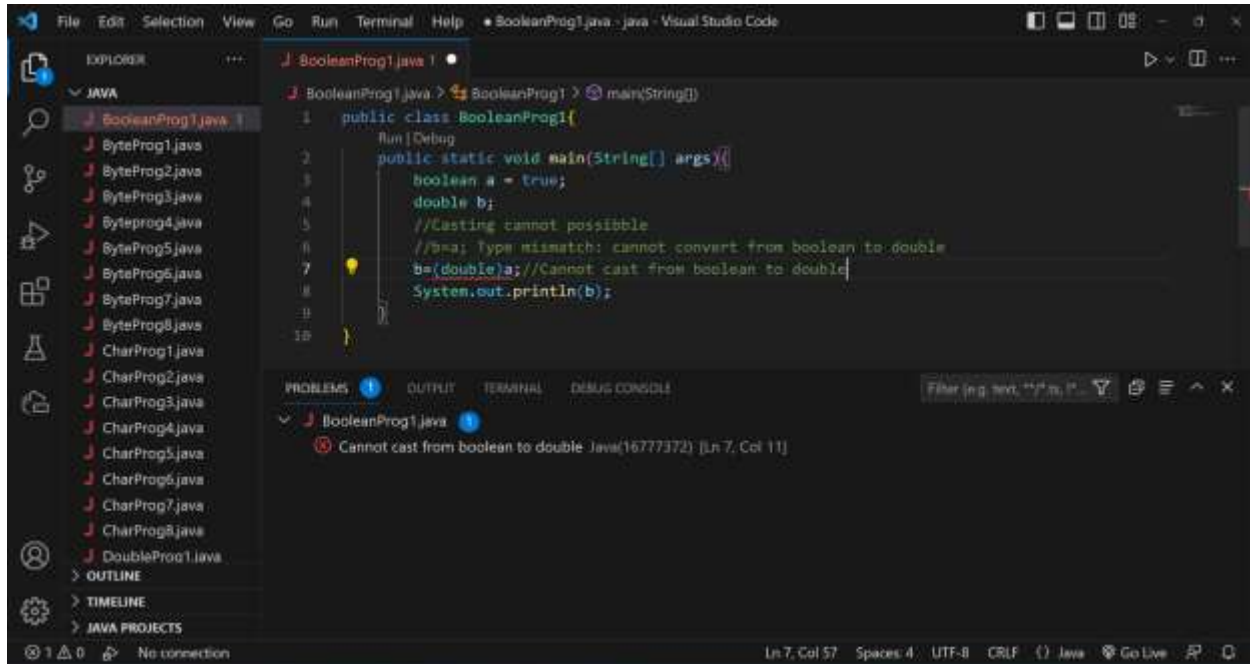


Program 63: (boolean to double)

```
public class BooleanProg1 {  
    public static void main(String[] args) {  
        boolean a = true;  
        double b;  
        //Casting cannot possible  
        //b=a; Type mismatch: cannot convert from boolean to double  
        b=(double)a; //Cannot cast from boolean to double  
        System.out.println(b);  
    }  
}
```

Casting Cannot Possible

Output:

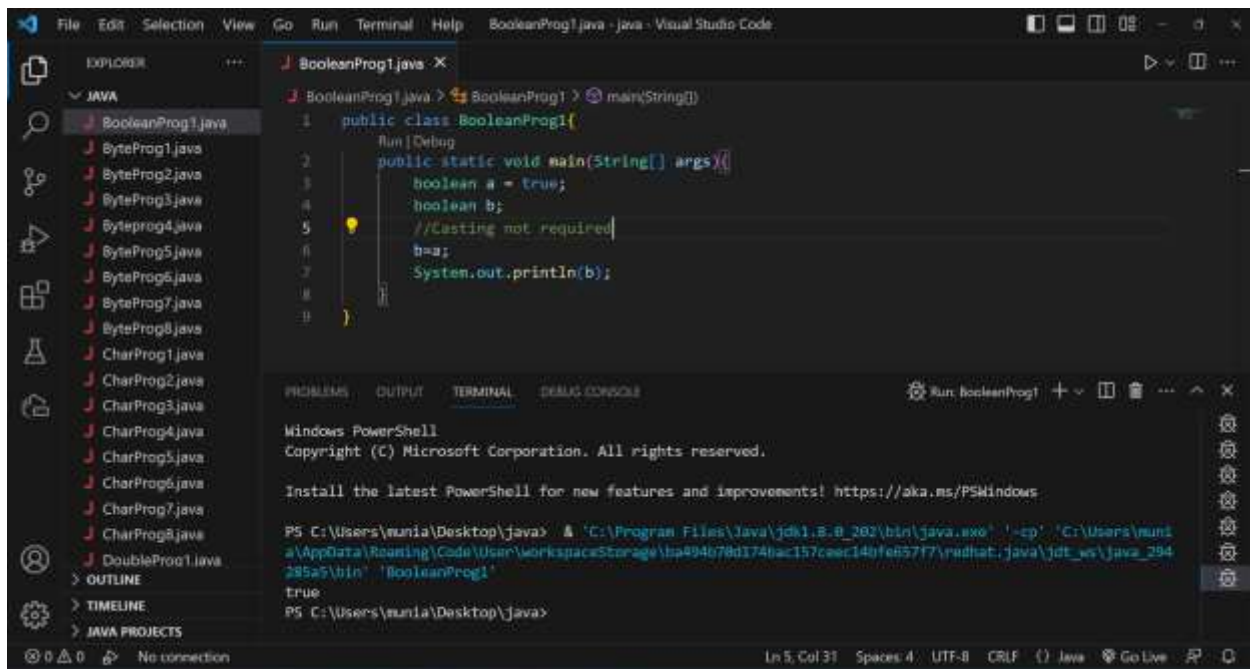


Program 64: (boolean to boolean)

```
public class BooleanProg1 {  
    public static void main(String[] args) {  
        boolean a = true;  
        boolean b;  
        //Casting not required  
        b=a;  
        System.out.println(b);  
    }  
}
```

Casting not required

Output:

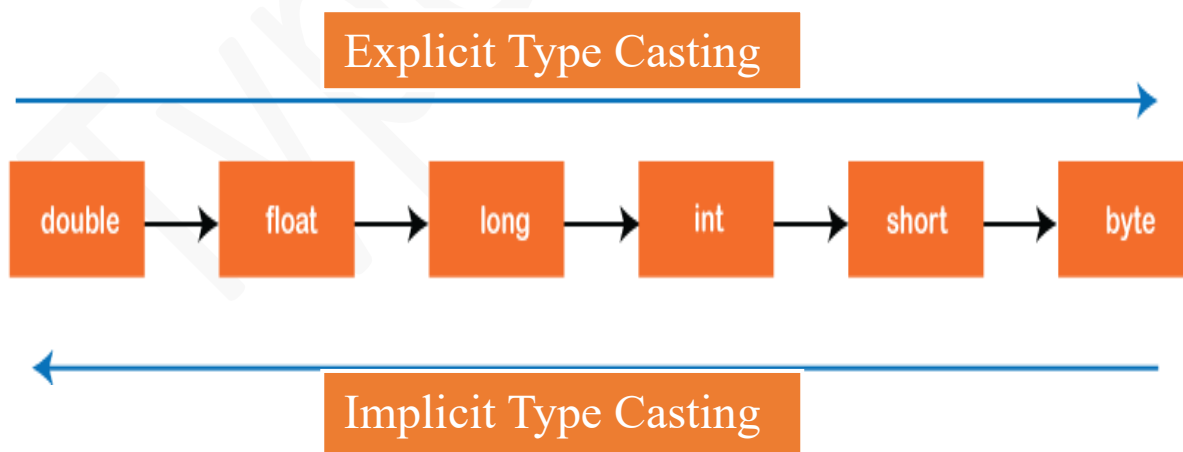


The screenshot shows the Visual Studio Code interface. The Explorer pane on the left lists several Java files, including BooleanProg1.java. The main editor displays the code for BooleanProg1.java, which is a public class with a main method. The code is as follows:

```
1 public class BooleanProg1 {  
2     public static void main(String[] args) {  
3         boolean a = true;  
4         boolean b;  
5         //Casting not required  
6         b=a;  
7         System.out.println(b);  
8     }  
9 }
```

The output pane at the bottom shows the execution of the program, displaying the output 'true'.

Type Casting Chart:



	char	byte	short	int	long	float	double	boolean
char	CNR	E	E	I	I	I	I	CNP
byte	E	CNR	I	I	I	I	I	CNP
short	E	E	CNR	I	I	I	I	CNP
int	E	E	E	CNR	I	I	I	CNP
long	E	E	E	E	CNR	I	I	CNP
float	E	E	E	E	E	CNR	I	CNP
double	E	E	E	E	E	E	CNR	CNP
boolean	CNP	CNP	CNP	CNP	CNP	CNP	CNP	CNR

Here,

CNR -- Casting Not Required

E-- Explicit Type Casting

I-- Implicit Type Casting

CNP-- Casting Not Possible