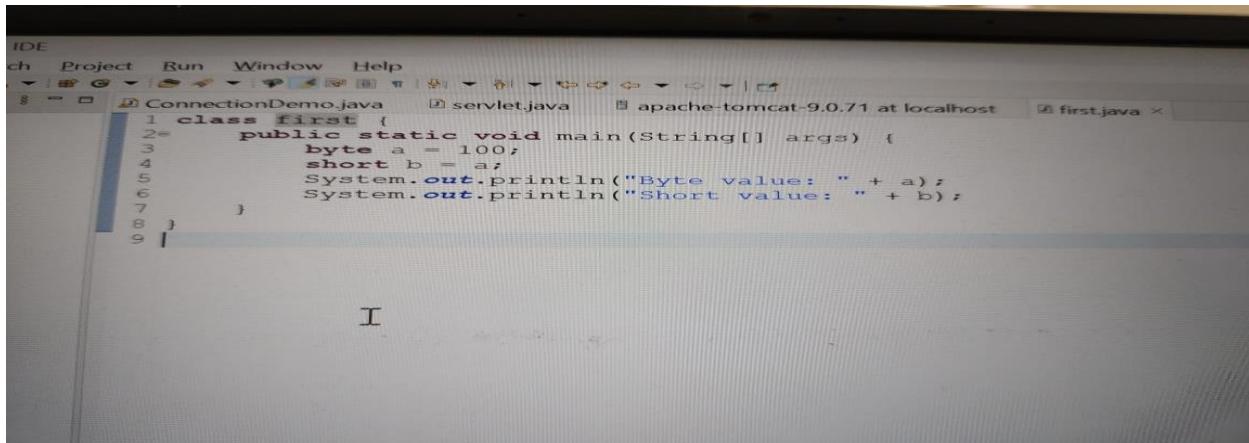


TYPE CASTING



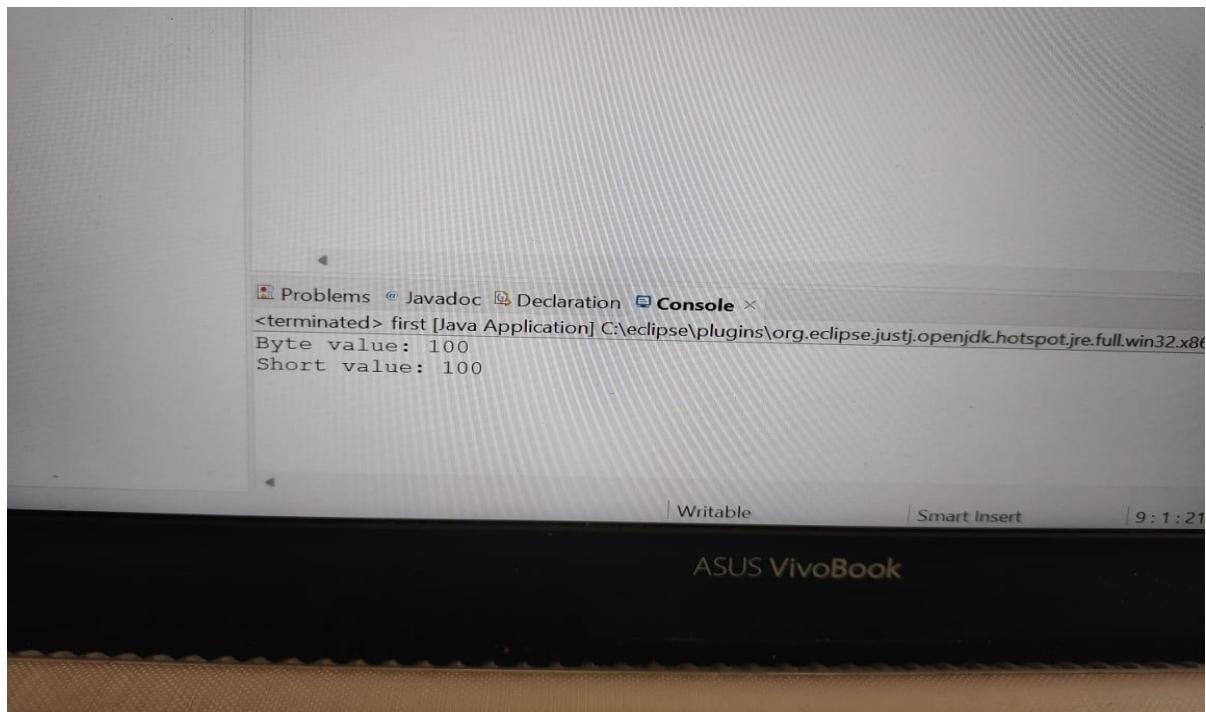
BY
EEMANI VIVEK

BYTE TO SHORT



The screenshot shows the Eclipse IDE interface. The menu bar includes 'IDE', 'Project', 'Run', 'Window', and 'Help'. The toolbar has various icons for file operations. The 'Project Explorer' view shows files like 'ConnectionDemo.java', 'servlet.java', 'apache-tomcat-9.0.71 at localhost', and 'first.java'. The 'Code Editor' displays the following Java code:

```
1 class first {
2     public static void main(String[] args) {
3         byte a = 100;
4         short b = a;
5         System.out.println("Byte value: " + a);
6         System.out.println("Short value: " + b);
7     }
8 }
```



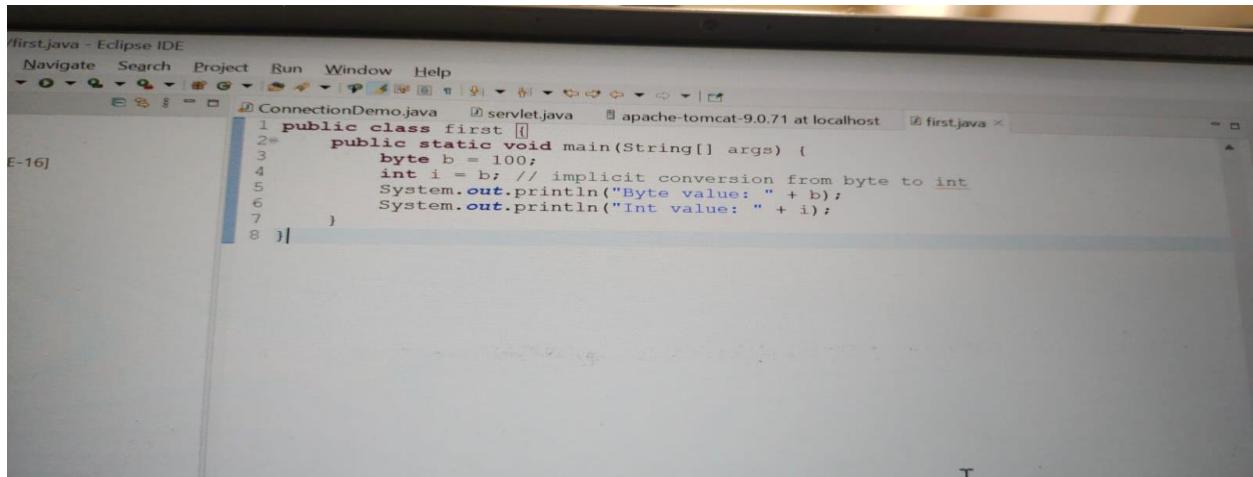
The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the results of the Java application's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64\bin\java.exe
Byte value: 100
Short value: 100
```

The bottom of the screen shows the laptop brand 'ASUS VivoBook'.

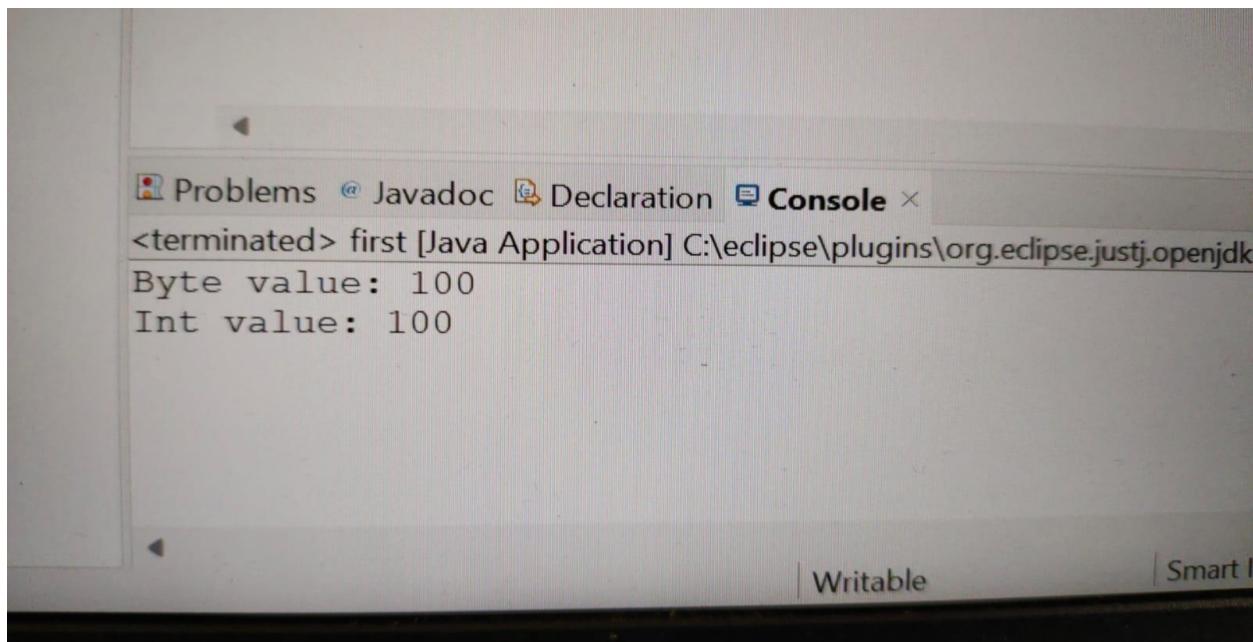
IMPLICIT CONVERSION

BYTE TO INT



A screenshot of the Eclipse IDE interface. The title bar says "first.java - Eclipse IDE". The menu bar includes "Navigate", "Search", "Project", "Run", "Window", and "Help". The toolbar has various icons for file operations. The left sidebar shows "E-16" and a tree view of project files. The main editor window displays the following Java code:

```
ConnectionDemo.java  servlet.java  apache-tomcat-9.0.71 at localhost  first.java ×  
E-16]  
1 public class first {  
2     public static void main(String[] args) {  
3         byte b = 100;  
4         int i = b; // implicit conversion from byte to int  
5         System.out.println("Byte value: " + b);  
6         System.out.println("Int value: " + i);  
7     }  
8 }
```



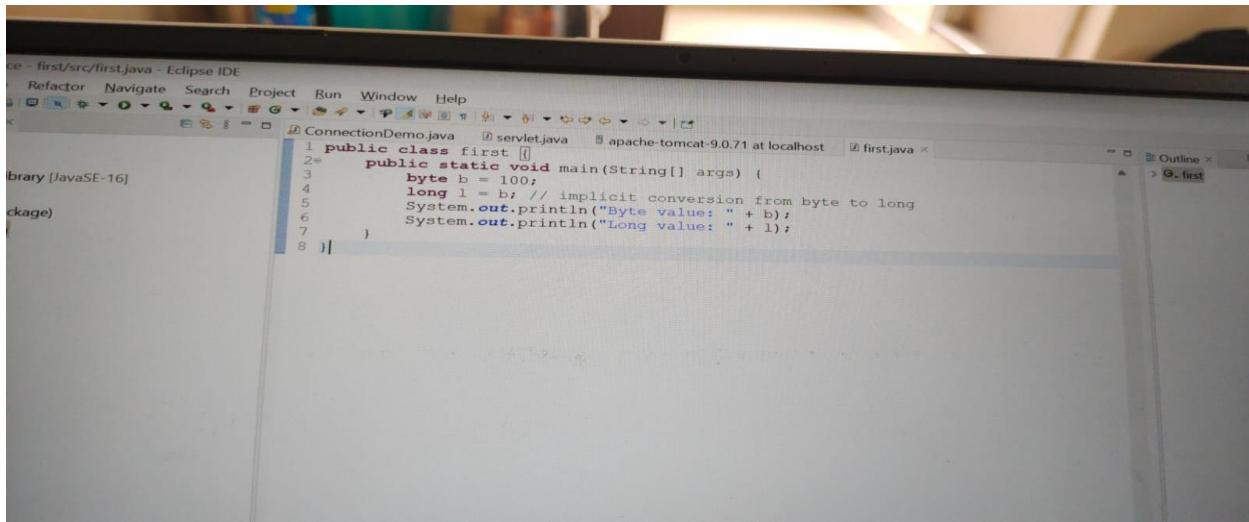
A screenshot of the Eclipse IDE interface focusing on the "Console" tab. The title bar says "first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk". The "Console" tab is selected, showing the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk  
Byte value: 100  
Int value: 100
```

The "Console" tab also has "Problems", "Javadoc", and "Declaration" tabs. At the bottom right, there are "Writable" and "Smart I" buttons.

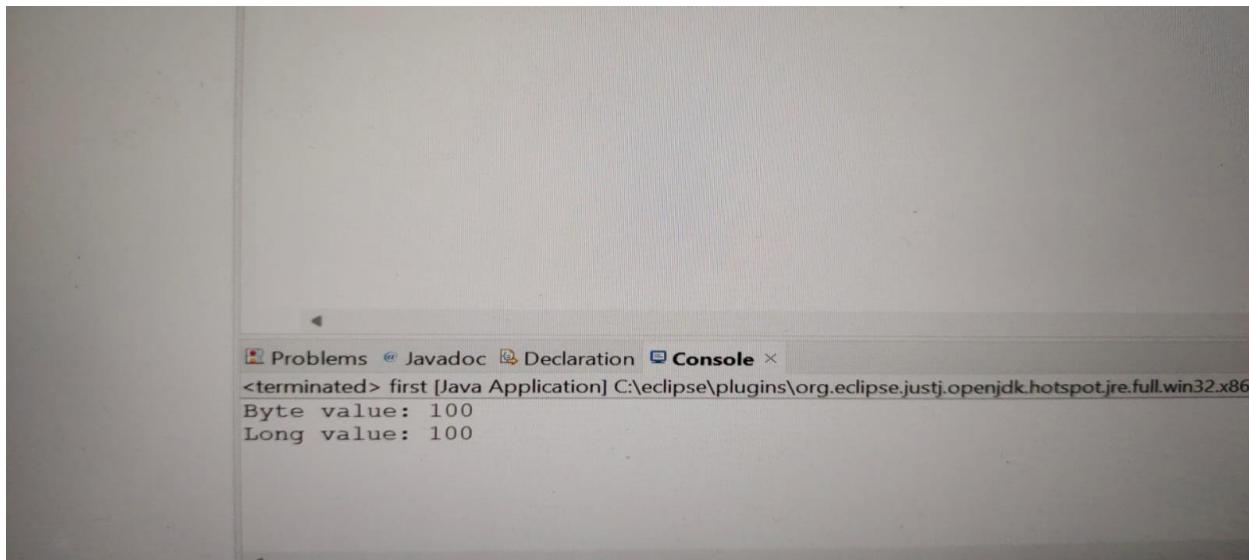
IMPLICIT CONVERSION

BYTE TO LONG



The screenshot shows the Eclipse IDE interface with a Java file named 'first.java' open. The code contains a main method that declares a byte variable 'b' and a long variable 'l'. It then prints the byte value and the long value to the console. The code is as follows:

```
1 public class first {
2     public static void main(String[] args) {
3         byte b = 100;
4         long l = b; // implicit conversion from byte to long
5         System.out.println("Byte value: " + b);
6         System.out.println("Long value: " + l);
7     }
8 }
```



The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The output window displays the results of the program execution. The byte value is printed as 100, and the long value is also printed as 100, demonstrating the implicit conversion from byte to long.

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86
Byte value: 100
Long value: 100
```

IMPLICIT CONVERSION

BYTE TO FLOAT

The screenshot shows the Eclipse IDE interface. At the top, the menu bar includes 'File', 'Factor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The central workspace displays a Java file named 'first.java' with the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         byte b = 100;
4         float f = b; // implicit conversion from byte to float
5         System.out.println("Byte value: " + b);
6         System.out.println("Float value: " + f);
7     }
8 }
```

Below the code editor is the 'Console' view, which shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.o
Byte value: 100
Float value: 100.0
```

The bottom right corner of the screen shows the 'ASUS VivoBook' logo.

IMPLICIT CONVERSION

BYTE TO DOUBLE

The screenshot shows the Eclipse IDE interface. The top menu bar includes 'Editor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. A toolbar with various icons is visible above the editor area. The left sidebar shows a project tree with 'JavaSE-16' selected. The main editor window displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         byte b = 100;
4         double f = b; // implicit conversion from byte to double
5         System.out.println("Byte value: " + b);
6         System.out.println("double value: " + f);
7     }
8 }
```

Below the editor is the 'Console' view, which shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wireframe
Byte value: 100
double value: 100.0
```

IMPLICIT CONVERSION

BYTE TO CHARACTER

The screenshot shows the Eclipse IDE interface. In the top center, there's a toolbar with various icons. Below it is a menu bar with 'File', 'Edit', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. The main workspace contains several tabs: 'ConnectionDemo.java', 'servlet.java', 'apache-tomcat-9.0.71 at localhost', and 'first.java'. The 'first.java' tab is active and displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         byte b = 100;
4         char c = (char) b; // explicit conversion from byte to char
5         System.out.println("Byte value: " + b);
6         System.out.println("Char value: " + c);
7     }
8 }
```

In the bottom right corner of the workspace, there's a small text 'JavaSE-16'.

Below the workspace is the 'Console' view, which shows the output of the application. It has tabs for 'Problems', '@ Javadoc', 'Declaration', and 'Console'. The 'Console' tab is selected and shows the following text:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.0.1.0.20210510-1200\java\bin\java.exe
Byte value: 100
Char value: d
```

The 'Console' view also has tabs for 'Problems', '@ Javadoc', 'Declaration', and 'Console'. The 'Console' tab is selected and shows the following text:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.0.1.0.20210510-1200\java\bin\java.exe
Byte value: 100
Char value: d
```

At the bottom of the screen, the laptop brand 'ASUS VivoBook' is visible.

EXPLICIT CONVERSION

BYTE TO BOOLEAN

The screenshot shows the Eclipse IDE interface. In the top center, there is a toolbar with various icons. Below the toolbar, the menu bar includes 'Refactor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. The main workspace contains several tabs: 'ConnectionDemo.java', 'servlet.java', 'apache-tomcat-9.0.71 at localhost', and 'first.java'. The 'first.java' tab is active and displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         byte b = 1;
4         boolean bool = (b != 0); // convert byte to boolean based on condition
5         System.out.println("Byte value: " + b);
6         System.out.println("Boolean value: " + bool);
7     }
8 }
```

Below the code editor is the 'Console' view, which shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.jdt.core\hotspot\jre\full\win32\x86
Byte value: 1
Boolean value: true
```

The bottom right corner of the screen shows the laptop brand 'ASUS VivoBook'.

EXPLICIT CONVERSION

SHORT TO BYTE

The screenshot shows the Eclipse IDE interface. The top window displays the code for `first.java`:

```
1 public class first {
2     public static void main(String[] args) {
3         short s = 1000;
4         byte b = (byte) s; // explicit conversion from short to byte
5         System.out.println("Short value: " + s);
6         System.out.println("Byte value: " + b);
7     }
8 }
```

The bottom window shows the `Console` tab with the following output:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.
Short value: 1000
Byte value: -24
```

A small portion of the screen is visible at the bottom right, showing the word "Writable".

EXPLICIT CONVERSION

SHORT TO INT

The screenshot shows the Eclipse IDE interface. The top part displays the code editor with a Java file named 'first.java'. The code defines a public class 'first' with a main method that prints the value of a short variable 's' and an int variable 'i' to the console. The bottom part shows the 'Console' view where the output of the application is displayed, showing the values 'Short value: 1000' and 'Int value: 1000'.

```
1 public class first {
2     public static void main(String[] args) {
3         short s = 1000;
4         int i = s; // implicit conversion from short to int
5         System.out.println("Short value: " + s);
6         System.out.println("Int value: " + i);
7     }
8 }
```

Problems Javadoc Declaration Console <terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot Short value: 1000 Int value: 1000 Writable Smart Insert

IMPLICIT CONVERSION

SHORT TO LONG

The screenshot shows the Eclipse IDE interface. In the top window, the code for `ConnectionDemo.java` is displayed:

```
1 public class first {
2     public static void main(String[] args) {
3         short s = 1000;
4         long i = s; // implicit conversion from short to long
5         System.out.println("Short value: " + s);
6         System.out.println("long value: " + i);
7     }
8 }
```

In the bottom window, the `Console` tab is selected, showing the output of the application:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.ecl
Short value: 1000
long value: 1000
```

The system tray at the bottom right indicates the computer is an **ASUS VivoB**.

IMPLICIT CONVERSION

SHORT TO FLOAT

The screenshot shows the Eclipse IDE interface. The top part displays the code for `first.java`:

```
1 public class first {
2     public static void main(String[] args) {
3         short s = 1000;
4         float i = s; // implicit conversion from short to float
5         System.out.println("Short value: " + s);
6         System.out.println("float value: " + i);
7     }
8 }
```

The bottom part shows the `Console` tab with the output:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86
Short value: 1000
float value: 1000.0
```

The laptop's branding 'ASUS VivoBook' is visible at the bottom.

IMPLICIT CONVERSION

SHORT TO DOUBLE

The screenshot shows the Eclipse IDE interface. The top window displays the Java code for a class named 'first'. The code defines a main method that initializes a short variable 's' to 1000 and then prints it as both a short and a double using System.out.println. The bottom window is the 'Console' view, which shows the output of the application's execution. The output text is:
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.
Short value: 1000
double value: 1000.0

```
1 public class first {  
2     public static void main(String[] args) {  
3         short s = 1000;  
4         double i = s; // implicit conversion from short to double  
5         System.out.println("Short value: " + s);  
6         System.out.println("double value: " + i);  
7     }  
8 }
```

Console output:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.  
Short value: 1000  
double value: 1000.0
```

IMPLICIT CONVERSION

SHORT TO CHARACTER

The screenshot shows the Eclipse IDE interface. At the top, the menu bar includes 'File', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The central workspace displays a Java file named 'first.java' with the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         short s = 1000;
4         char b = (char) s; // explicit conversion from short to char
5         System.out.println("Short value: " + s);
6         System.out.println("char value: " + b);
7     }
8 }
```

Below the code editor is the 'Console' view, which shows the application's output:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.o
Short value: 1000
char value: ?
```

The bottom right corner of the screen shows the 'ASUS VivoBook' logo.

EXPLICIT CONVERSION

SHORT TO BOOLEAN

The screenshot shows the Eclipse IDE interface. The top part displays the code for a Java class named 'first' in the file 'ConnectionDemo.java'. The code defines a main method that initializes a short variable 's' to 10, then converts it to a boolean value 'bool' using the condition `(s != 0)`. It prints both the short value and the boolean value to the console. The bottom part shows the 'Console' view with the output:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
Short value: 10
Boolean value: true
```

The Eclipse interface includes a toolbar, a menu bar with options like Navigate, Search, Project, Run, Window, Help, and a status bar at the bottom.

EXPLICIT CONVERSION

INT TO BYTE

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Search, Project, Run, Window, and Help. The toolbar contains various icons for file operations. The left sidebar lists ConnectionDemo.java, servlet.java, apache-tomcat-9.0.71 at localhost, and first.java. The main editor window displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         int i = 1000;
4         byte b = (byte) i; // explicit conversion from int to byte
5         System.out.println("Int value: " + i);
6         System.out.println("Byte value: " + b);
7     }
8 }
```

The bottom half of the screen shows the Eclipse Console view. It displays the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk
Int value: 1000
Byte value: -24
```

The ASUS VivoBook logo is visible at the bottom of the physical device.

EXPLICIT CONVERSION

INT TO SHORT

The screenshot shows the Eclipse IDE interface. In the top window, a Java file named 'first.java' is open, containing the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         int i = 1000;
4         short b = (short) i; // explicit conversion from int to short
5         System.out.println("Int value: " + i);
6         System.out.println("short value: " + b);
7     }
8 }
```

In the bottom window, the 'Console' tab is selected, showing the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspace
Int value: 1000
short value: 1000
```

The laptop's brand, 'ASUS VivoBook', is visible at the bottom of the screen.

EXPLICIT CONVERSION

INT TO LONG

The screenshot shows the Eclipse IDE interface. At the top, the menu bar includes 'Editor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The left sidebar displays 'JavaSE-16' under 'Projects'. The main workspace shows a Java file named 'first.java' with the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         int i = 1000;
4         long l = i; // implicit conversion from int to long
5         System.out.println("Int value: " + i);
6         System.out.println("Long value: " + l);
7     }
8 }
```

At the bottom, the 'Console' tab is selected in the 'Problems' view. The output window shows the execution results:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
Int value: 1000
Long value: 1000
```

IMPLICIT CONVERSION

INT TO FLOAT

The screenshot shows the Eclipse IDE interface. The top menu bar includes 'File', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. The toolbar below has various icons for file operations. The left sidebar shows a project structure with 'src/first.java - Eclipse IDE' and '[JavaSE-16]'. The main editor window displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         int i = 1000;
4         float l = i; // implicit conversion from int to float
5         System.out.println("Int value: " + i);
6         System.out.println("float value: " + l);
7     }
8 }
```

The code defines a class named 'first' with a main method. It declares an integer variable 'i' with the value 1000 and a float variable 'l' by assigning it the value of 'i'. Both are then printed to the console using System.out.println. The output is visible in the 'Console' tab at the bottom:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Int value: 1000
float value: 1000.0
```

The bottom status bar indicates the text is 'Writable' and shows the 'Smart Insert' icon. The laptop brand 'ASUS VivoBook' is visible at the bottom.

IMPLICIT CONVERSION

INT TO DOUBLE

The screenshot shows the Eclipse IDE interface. The top part displays the code for a Java class named 'first' in the 'first.java' file. The code initializes an integer variable 'i' to 1000 and then converts it to a double variable 'l'. The output of this conversion is printed to the console. The bottom part of the screenshot shows the Eclipse Console view, which displays the output: 'Int value: 1000' and 'double value: 1000.0'. The laptop's brand name, 'ASUS VivoBook', is visible at the bottom of the screen.

```
1 public class first {
2     public static void main(String[] args) {
3         int i = 1000;
4         double l = i; // implicit conversion from int to double
5         System.out.println("Int value: " + i);
6         System.out.println("double value: " + l);
7     }
8 }
```

Problems Javadoc Declaration Console <terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Int value: 1000
double value: 1000.0

Writable Smart Insert

ASUS VivoBook

IMPLICIT CONVERSION

INT TO CHARACTER

The screenshot shows the Eclipse IDE interface. At the top, the menu bar includes Refactor, Navigate, Search, Project, Run, Window, and Help. Below the menu is a toolbar with various icons. The central workspace displays a Java file named 'first.java' with the following code:

```
1 public class first {  
2     public static void main(String[] args) {  
3         int i = 65;  
4         char c = (char) i; // explicit conversion from int to char  
5         System.out.println("Int value: " + i);  
6         System.out.println("Char value: " + c);  
7     }  
8 }
```

Below the code editor is the 'Console' view, which shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre_int  
Int value: 65  
Char value: A
```

The bottom right corner of the screen shows the 'ASUS VivoBook' logo.

EXPLICIT CONVERSION

INT TO BOOLEAN

The screenshot shows the Eclipse IDE interface with the title bar "first.java - Eclipse IDE". The menu bar includes "Navigate", "Search", "Project", "Run", "Window", and "Help". The toolbar has various icons for file operations like new, open, save, cut, copy, paste, etc. The left sidebar shows a project tree with "E-16" selected. The main editor window displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         int i = 0;
4         boolean bool = (i != 0); // convert int to boolean based on condition
5         System.out.println("Int value: " + i);
6         System.out.println("Boolean value: " + bool);
7     }
8 }
```

Below the editor is the "Console" view, which shows the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hc
Int value: 0
Boolean value: false
```

The status bar at the bottom right indicates "Writable" and "Smart Insert".

EXPLICIT CONVERSION

LONG TO BYTE

The screenshot shows the Eclipse IDE interface. The top menu bar includes 'File', 'Edit', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The left sidebar shows project files: 'ConnectionDemo.java', 'servlet.java', 'apache-tomcat-9.0.71 at localhost', and 'first.java'. The main editor window displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         long l = 1000L;
4         byte b = (byte) l; // explicit conversion from long to byte
5         System.out.println("Long value: " + l);
6         System.out.println("Byte value: " + b);
7     }
8 }
```

The code uses an explicit cast ((byte) l) to convert the long variable l to a byte variable b. The output window at the bottom shows the results of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64
Long value: 1000
Byte value: -24
```

The status bar at the bottom right indicates the time as 1:19:18.

EXPLICIT CONVERSION

LONG TO SHORT

The screenshot shows the Eclipse IDE interface. In the top window, the code for `first.java` is displayed:

```
1 public class first {  
2     public static void main(String[] args) {  
3         long l = 1000L;  
4         short b = (short) l; // explicit conversion from long to short  
5         System.out.println("Long value: " + l);  
6         System.out.println("Short value: " + b);  
7     }  
8 }
```

In the bottom window, the `Console` tab shows the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v2021-06-15T12-05-01Z  
Long value: 1000  
short value: 1000
```

The laptop's brand, **ASUS VivoBook**, is visible at the bottom of the screen.

EXPLICIT CONVERSION

LONG TO INT

The screenshot shows the Eclipse IDE interface. The top part displays the code for a Java class named 'first'. The code initializes a long variable 'l' to 1000L and then performs an explicit conversion to an int variable 'b' using the cast operator (int). It then prints both values to the standard output. The bottom part shows the 'Console' view where the output is displayed: 'Long value: 1000' and 'int value: 1000'. The laptop's brand name, 'ASUS VivoBook', is visible at the bottom.

```
first/src/first.java - Eclipse IDE
Refactor Navigate Search Project Run Window Help
ConnectionDemo.java servlet.java apache-tomcat-9.0.71 at localhost *first.java ×
ary [JavaSE-16]
age)
1 public class first {
2     public static void main(String[] args) {
3         long l = 1000L;
4         int b = (int) l; // explicit conversion from long to int
5         System.out.println("Long value: " + l);
6         System.out.println("int value: " + b);
7     }
8 }
```

Problems Javadoc Declaration Console ×
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot
Long value: 1000
int value: 1000

Writable Smart Insert

ASUS VivoBook

EXPLICIT CONVERSION

LONG TO FLOAT

The screenshot shows the Eclipse IDE interface. The top menu bar includes Navigate, Search, Project, Run, Window, and Help. The toolbar below has various icons for file operations like Open, Save, Cut, Copy, Paste, and Find. The left sidebar shows project files: ConnectionDemo.java, servlet.java, apache-tomcat-9.0.71 at localhost, and first.java. The main editor window displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         long l = 1000000000L;
4         float f = l; // implicit conversion from long to float
5         System.out.println("Long value: " + l);
6         System.out.println("Float value: " + f);
7     }
8 }
```

The bottom right panel is the Console tab, which shows the output of the application's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.o
Long value: 1000000000
Float value: 1.0E9
```

The status bar at the bottom right indicates the text is Writable.

IMPLICIT CONVERSION

LONG TO DOUBLE

The screenshot shows the Eclipse IDE interface. The top window displays the code for `first.java`:

```
1 public class first {  
2     public static void main(String[] args) {  
3         long l = 1000000000L;  
4         double f = l; // implicit conversion from long to double  
5         System.out.println("Long value: " + l);  
6         System.out.println("double value: " + f);  
7     }  
8 }
```

The bottom window is the Console view, showing the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.edip  
Long value: 1000000000  
double value: 1.0E9
```

The system tray at the bottom right shows the ASUS Vivo logo.

IMPLICIT CONVERSION

LONG TO CHARACTER

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows "first.java - Eclipse IDE" and a menu bar with "Navigate", "Search", "Project", "Run", "Window", and "Help".
- Project Explorer:** Shows a project named "SE-16" containing files like "ConnectionDemo.java", "servlet.java", "apache-tomcat-9.0.71 at localhost", and "first.java".
- Code Editor:** Displays the following Java code:

```
public class first {
    public static void main(String[] args) {
        long l = 10000000000L;
        char c = (char) l; // explicit conversion from long to char
        System.out.println("Long value: " + l);
        System.out.println("Char value: " + c);
    }
}
```
- Console View:** Shows the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.op
Long value: 10000000000
Char value: ?
```
- Bottom Status Bar:** Shows "Writable" and "Sn".
- Laptop Brand:** The laptop is an ASUS VivoBook.

EXPLICIT CONVERSION

LONG TO BOOLEAN

The screenshot shows the Eclipse IDE interface. At the top, the menu bar includes 'File', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The 'Project Explorer' view on the left shows files like 'ConnectionDemo.java', 'servlet.java', 'apache-tomcat-9.0.71 at localhost', and 'first.java'. The main editor area displays the following Java code:

```
1 public class first {  
2     public static void main(String[] args) {  
3         long l = 0L;  
4         boolean b = (l != 0L); // convert long to boolean based on condition  
5         System.out.println("Long value: " + l);  
6         System.out.println("Boolean value: " + b);  
7     }  
8 }
```

Below the editor is the 'Console' view, which shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jdk11  
Long value: 0  
Boolean value: false
```

EXPLICIT CONVERSION

FLOAT TO BYTE

The screenshot shows the Eclipse IDE interface. The top menu bar includes 'File', 'Edit', 'Search', 'Project', 'Run', 'Window', and 'Help'. The toolbar below has various icons for file operations. The central workspace shows a Java file named 'first.java' with the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         float f = 123.45f;
4         byte b = (byte) f; // explicit conversion from float to byte
5         System.out.println("Float value: " + f);
6         System.out.println("Byte value: " + b);
7     }
8 }
```

Below the code editor is the 'Console' view, which displays the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.java.11.jdk\lib\jrt-fs.jar
Float value: 123.45
Byte value: 123
```

The bottom right corner of the screen shows the 'ASUS VivoBook' logo.

EXPLICIT CONVERSION

FLOAT TO SHORT

The screenshot shows the Eclipse IDE interface. The top part displays the code for a Java class named 'first' in a file named 'first.java'. The code demonstrates an explicit conversion from a float to a short:

```
1 public class first {
2     public static void main(String[] args) {
3         float f = 123.45f;
4         short b = (short) f; // explicit conversion from float to short
5         System.out.println("Float value: " + f);
6         System.out.println("short value: " + b);
7     }
8 }
```

The bottom part shows the 'Console' tab with the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.jdt.core\openjdk.hotspot.jvm
Float value: 123.45
short value: 123
```

The laptop's brand name, 'ASUS VivoBook', is visible at the bottom of the screen.

EXPLICIT CONVERSION

FLOAT TO INT

The screenshot shows the Eclipse IDE interface. The top window displays the code for a Java class named 'first'. The code contains a main method that prints the float value 123.45f and its integer representation 123. The bottom window, titled 'Console', shows the output of the application, which includes the printed float and integer values.

```
src/first.java - Eclipse IDE
File Navigate Search Project Run Window Help
ConnectionDemo.java servlet.java apache-tomcat-9.0.71 at localhost first.java
1 public class first {
2     public static void main(String[] args) {
3         float f = 123.45f;
4         int b = (int) f; // explicit conversion from float to int
5         System.out.println("Float value: " + f);
6         System.out.println("int value: " + b);
7     }
8 }
```

```
Problems Javadoc Declaration Console <terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.open_Float value: 123.45
int value: 123
```

EXPLICIT CONVERSION

FLOAT TO LONG

The screenshot shows the Eclipse IDE interface. The top menu bar includes Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The left sidebar displays the Project Explorer, showing a Java project named 'first' with files like ConnectionDemo.java, servlet.java, apache-tomcat-9.0.71 at localhost, and first.java. The first.java file is open in the editor, containing the following code:

```
1 public class first {  
2     public static void main(String[] args) {  
3         float f = 123.45f;  
4         long b = (long) f; // explicit conversion from float to long  
5         System.out.println("Float value: " + f);  
6         System.out.println("long value: " + b);  
7     }  
8 }
```

The bottom right panel is the Console, which shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hots  
Float value: 123.45  
long value: 123
```

The status bar at the bottom indicates the text is Writable and features Smart Insert. The laptop brand 'ASUS VivoBook' is visible at the bottom.

EXPLICIT CONVERSION

FLOAT TO DOUBLE

The screenshot shows the Eclipse IDE interface. In the top center, there's a toolbar with icons for file operations like New, Open, Save, and Print. Below the toolbar is a menu bar with 'File', 'Edit', 'Search', 'Project', 'Run', 'Window', and 'Help'. The main workspace displays a Java code editor with the file 'first.java' open. The code is as follows:

```
1 public class first{
2     public static void main(String[] args) {
3         float f = 123.45f;
4         double d = f; // implicit conversion from float to double
5         System.out.println("Float value: " + f);
6         System.out.println("Double value: " + d);
7     }
8 }
```

Below the code editor is a 'Console' view window. It shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Float value: 123.45
Double value: 123.44999694824219
```

The bottom right corner of the screen shows the 'ASUS VivoBook' logo.

IMPLICIT CONVERSION

FLOAT TO CHARACTER

The screenshot shows the Eclipse IDE interface. The top window displays the Java code for a class named 'first'. The code defines a main method that creates a float variable 'f' with a value of 65.0, converts it to a character 'c' using the cast operator (char) f;, and then prints both the float and character values to the console. The bottom window shows the 'Console' tab with the output of the program, which includes the float value 65.0 and the character value A.

```
1 public class first {  
2     public static void main(String[] args) {  
3         float f = 65.0f; |  
4         char c = (char) f; // explicit conversion from float to char  
5         System.out.println("Float value: " + f);  
6         System.out.println("Char value: " + c);  
7     }  
8 }
```

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.  
Float value: 65.0  
Char value: A
```

EXPLICIT CONVERSION

FLOAT TO BOOLEAN

The screenshot shows the Eclipse IDE interface. The top menu bar includes Factor, Navigate, Search, Project, Run, Window, and Help. The toolbar below has various icons for file operations. The main workspace displays a Java file named 'first.java' with the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         float f = 0.0f;
4         boolean b = (f != 0.0f); // convert float to boolean based on condition
5         System.out.println("Float value: " + f);
6         System.out.println("Boolean value: " + b);
7     }
8 }
```

Below the editor is the 'Console' view, which shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.jdt.core\1.6.0\jre\bin\javaw.exe
Float value: 0.0
Boolean value: false
```

The bottom right corner of the screen shows the 'ASUS VivoBook' logo.

EXPLICIT CONVERSION

DOUBLE TO BYTE

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows the title "/first.java - Eclipse IDE" and a menu bar with options: Navigate, Search, Project, Run, Window, Help.
- Toolbar:** Standard Eclipse toolbar icons.
- Project Explorer:** Shows a project named "SE-16".
- Code Editor:** Displays the following Java code in the file "first.java":

```
1 public class first {
2     public static void main(String[] args) {
3         double d = 123.45;
4         byte b = (byte) d; // explicit conversion from double to byte
5         System.out.println("Double value: " + d);
6         System.out.println("Byte value: " + b);
7     }
8 }
```

- Console View:** Shows the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspo
Double value: 123.45
Byte value: 123
```

- Bottom Status Bar:** Shows "Writable" and "Smart Insert" status.
- Laptop Brand:** The laptop is an ASUS VivoBook.

EXPLICIT CONVERSION

DOUBLE TO SHORT

The screenshot shows the Eclipse IDE interface. In the top center, the title bar reads "e-workspace - first/src/first.java - Eclipse IDE". Below the title bar is the menu bar with options like File, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The "File" menu is currently highlighted.

In the center-left area, the "Editor" tab is active, displaying the Java code for "first.java". The code is as follows:

```
1 public class first {  
2     public static void main(String[] args) {  
3         double d = 123.45;  
4         short b = (short) d; // explicit conversion from double to short  
5         System.out.println("Double value: " + d);  
6         System.out.println("short value: " + b);  
7     }  
8 }
```

To the right of the editor, there are several toolbars and panes. A toolbar with icons for file operations, search, and run is visible. To the right of the toolbar is the "Outline" view, which shows the class structure. Below the Outline is the "Problems" view, which is currently empty. At the bottom of the screen, the status bar displays "Writable" and "Smart Insert".

In the bottom right corner of the screen, the text "ASUS VivoBook" is visible, indicating the brand of the laptop.

The bottom half of the image shows the physical keyboard and trackpad of the ASUS VivoBook laptop.

EXPLICIT CONVERSION

DOUBLE TO INT

The screenshot shows the Eclipse IDE interface. At the top, the menu bar includes 'Editor', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The central workspace displays a Java file named 'first.java' with the following code:

```
1 public class first {  
2     public static void main(String[] args) {  
3         double d = 123.45;  
4         int b = (int) d; // explicit conversion from double to int  
5         System.out.println("Double value: " + d);  
6         System.out.println("int value: " + b);  
7     }  
8 }
```

Below the code editor is a 'Console' tab. The output window shows the results of running the application:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.op  
Double value: 123.45  
int value: 123
```

The bottom right corner of the screen shows the 'ASUS VivoBook' logo.

EXPLICIT CONVERSION

DOUBLE TO LONG

The screenshot shows the Eclipse IDE interface. The top window displays the Java code for a class named 'first'. The code defines a main method that initializes a double variable 'd' to 123.45, performs an explicit conversion to long using the cast operator, and prints both the original double value and the converted long value to the standard output.

```
1 public class first {
2     public static void main(String[] args) {
3         double d = 123.45;
4         long b = (long) d; // explicit conversion from double to long
5         System.out.println("Double value: " + d);
6         System.out.println("long value: " + b);
7     }
8 }
```

The bottom window is the Console view, showing the output of the program. It displays the string "Double value: 123.45" followed by "long value: 123".

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.l
Double value: 123.45
long value: 123
```

The laptop's brand name, "ASUS VivoBook", is visible at the bottom of the screen.

EXPLICIT CONVERSION

DOUBLE TO CHARACTER

The screenshot shows the Eclipse IDE interface. The top part displays the code for a Java class named 'first' in the 'src' folder of a project named 'first'. The code performs an explicit conversion from a double value to a character and prints both values. The bottom part shows the 'Console' view where the output is displayed.

```
workspace - first/src/first.java - Eclipse IDE
Source Refactor Navigate Search Project Run Window Help
Explorer X ConnectionDemo.java servlet.java apache-tomcat-9.0.71 at localhost first.java X Outlines
dbcDemo
System Library [JavaSE-16]
default package)
first.java
va
a

1 public class first {
2     public static void main(String[] args) {
3         double d = 65.0;
4         char c = (char) d; // explicit conversion from double to char
5         System.out.println("Double value: " + d);
6         System.out.println("Char value: " + c);
7     }
8 }
```

Problems Javadoc Declaration Console X
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.war
Double value: 65.0
Char value: A

Writable Smart Insert

ASUS VivoBook

EXPLICIT CONVERSION

DOUBLE TO BOOLEAN

The screenshot shows the Eclipse IDE interface. At the top, the menu bar includes 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The central workspace displays three tabs: 'ConnectionDemo.java', 'servlet.java', and 'first.java'. The 'first.java' tab contains the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         double d = 0.0;
4         boolean b = (d != 0.0); // convert double to boolean based on condition
5         System.out.println("Double value: " + d);
6         System.out.println("Boolean value: " + b);
7     }
8 }
```

Below the code editor is the 'Console' view, which shows the output of the application's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.jdt.core\src
Double value: 0.0
Boolean value: false
```

The bottom right corner of the screen shows the 'ASUS VivoBook' logo.

EXPLICIT CONVERSION

CHARACTER TO BYTE

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** first/src/first.java - Eclipse IDE, Refactor, Navigate, Search, Project, Run, Window, Help.
- Project Explorer:** JavaSE-16
- Code Editor:** first.java (selected tab) contains the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         char c = 'A';
4         byte b = (byte) c; // explicit conversion from char to byte
5         System.out.println("Char value: " + c);
6         System.out.println("Byte value: " + b);
7     }
8 }
```
- Console Output:** Shows the execution results:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot\jre\bin\java
Char value: A
Byte value: 65
```
- Bottom Status Bar:** Writable, Smart Insert
- Bottom Branding:** ASUS VivoBook

IMPLICIT CONVERSION

CHARACTER TO SHORT

The screenshot shows the Eclipse IDE interface. In the top center, the title bar reads "st/src/first.java - Eclipse IDE". Below the title bar is the menu bar with options: Actor, Navigate, Search, Project, Run, Window, Help. The toolbar below the menu bar contains various icons for file operations like Open, Save, Cut, Copy, Paste, etc. The main workspace shows a Java file named "first.java" with the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         char c = 'A';
4         short b = (short) c; // explicit conversion from char to short
5         System.out.println("Char value: " + c);
6         System.out.println("short value: " + b);
7     }
8 }
```

Below the workspace is the "Console" view, which displays the output of the program's execution:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre
Char value: A
short value: 65
```

The bottom right corner of the screen shows the "ASUS VivoBook" logo.

IMPLICIT CONVERSION

CHARACTER TO INT

The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Navigate, Search, Project, Run, Window, Help. The title bar says "/src/first.java - Eclipse IDE". The left sidebar shows "JavaSE-16" and a package tree. The main editor window displays the following Java code:

```
1 public class first {  
2     public static void main(String[] args) {  
3         char c = 'A';  
4         int b = (int) c; // explicit conversion from char to int  
5         System.out.println("Char value: " + c);  
6         System.out.println("int value: " + b);  
7     }  
8 }
```

Below the editor is the "Console" view, which shows the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.  
Char value: A  
int value: 65
```

The status bar at the bottom right indicates "Writable".

IMPLICIT CONVERSION

CHARACTER TO FLOAT

The screenshot shows the Eclipse IDE interface. The top menu bar includes Refactor, Navigate, Search, Project, Run, Window, and Help. The toolbar has various icons for file operations. The left sidebar shows a Library [JavaSE-16] and package structure. The main editor window displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         char c = 'A';
4         float b = (float) c; // explicit conversion from char to float
5         System.out.println("Char value: " + c);
6         System.out.println("float value: " + b);
7     }
8 }
```

The bottom right corner of the editor shows the text "ASUS VivoBook". The bottom status bar indicates the code is Writable and has Smart Insert enabled.

In the bottom right panel, titled "Console", the output of the program is shown:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.w
Char value: A
float value: 65.0
```

IMPLICIT CONVERSION

CHARACTER TO BOOLEAN

The screenshot shows the Eclipse IDE interface. The top menu bar includes 'File', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The left sidebar shows a project structure with 'vaSE-16' expanded. The main editor window displays the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         char c = 'A';
4         double b = (double) c; // explicit conversion from char to double
5         System.out.println("Char value: " + c);
6         System.out.println("Double value: " + b);
7     }
8 }
```

The code uses an explicit conversion from a character 'A' to a double 'b'. The output window at the bottom shows the results of the program's execution:

```
Char value: A
Boolean value: true
```

The bottom right corner of the screen shows the text 'ASUS VivoBook'.

EXPLICIT CONVERSION

BOOLEAN TO BYTE

The screenshot shows the Eclipse IDE interface. In the top editor area, a Java file named 'first.java' is open, containing the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         boolean b = true;
4         byte by = (byte) (b ? 1 : 0); // ternary operator to convert boolean to byte
5         System.out.println("Boolean value: " + b);
6         System.out.println("Byte value: " + by);
7     }
8 }
```

In the bottom 'Console' tab, the output of the program is displayed:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.jdt.core\openjdk
Boolean value: true
Byte value: 1
```

The laptop's screen is visible in the background, showing the text 'ASUS VivoBook'.

EXPLICIT CONVERSION

BOOLEAN TO SHORT

The screenshot shows the Eclipse IDE interface. The top menu bar includes 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. Below the menu is a toolbar with various icons. The left sidebar shows project files: 'ConnectionDemo.java', 'servlet.java', 'apache-tomcat-9.0.71 at localhost', and 'first.java'. The main editor window displays the following Java code:

```
SE-16]
1 public class first {
2     public static void main(String[] args) {
3         boolean b = true;
4         short by = (short) (b ? 1 : 0); // ternary operator to convert boolean to
5         System.out.println("Boolean value: " + b);
6         System.out.println("short value: " + by);
7     }
8 }
```

The code uses a ternary operator to convert a boolean value to a short value. The output of the program is visible in the 'Console' tab below:

```
Problems Javadoc Declaration Console <terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1
Boolean value: true
short value: 1
```

The bottom of the screen shows the ASUS VivoBook laptop branding.

EXPLICIT CONVERSION

BOOLEAN TO INT

The screenshot shows an Eclipse IDE window on an ASUS VivoBook laptop. The code editor displays a Java file named ConnectionDemo.java with the following content:

```
1 public class first {
2     public static void main(String[] args) {
3         boolean b = true;
4         int by = (int) (b ? 1 : 0); // ternary operator to convert boolean to int
5         System.out.println("Boolean value: " + b);
6         System.out.println("int value: " + by);
7     }
8 }
```

The console output shows the execution results:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin\javaw.exe [2]
Boolean value: true
int value: 1
```

The laptop's keyboard is visible at the bottom of the screen.

EXPLICIT CONVERSION

BOOLEAN TO LONG

The screenshot shows the Eclipse IDE interface. In the top center, there's a toolbar with various icons. Below it is a menu bar with 'File', 'Edit', 'Navigate', 'Search', 'Project', 'Run', 'Window', and 'Help'. The main workspace contains several tabs: 'ConnectionDemo.java', 'servlet.java', 'apache-tomcat-9.0.71 at localhost', and 'first.java'. The 'first.java' tab is active, displaying the following Java code:

```
1 public class first {
2     public static void main(String[] args) {
3         boolean b = true;
4         float by = (float) (b ? 1 : 0); // ternary operator to convert boolean to
5         System.out.println("Boolean value: " + b);
6         System.out.println("float value: " + by);
7     }
8 }
```

In the bottom right corner of the workspace, there's a small preview of the Java code. Below the workspace is a 'Console' view tab. The 'Console' tab is active and shows the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1
Boolean value: true
float value: 1.0
```

At the bottom of the screen, there's a dark bar with the text 'ASUS VivoBook' and some status indicators: 'Writable', 'Smart Insert', and '6:24:259'.

EXPLICIT CONVERSION

BOOLEAN TO DOUBLE

The screenshot shows the Eclipse IDE interface. In the top left, the title bar says "src/first.java - Eclipse IDE". The menu bar includes "File", "Navigate", "Search", "Project", "Run", "Window", and "Help". Below the menu is a toolbar with various icons. The central workspace shows a Java file named "first.java" with the following code:

```
1 public class first {
2     public static void main(String[] args) {
3         boolean b = true;
4         double by = (double) (b ? 1 : 0); // ternary operator to convert boolean to double
5         System.out.println("Boolean value: " + b);
6         System.out.println("double value: " + by);
7     }
8 }
```

To the right of the code editor is the "Outline" view, which shows the class "first" and its method "main". At the bottom of the screen is the "Console" view, which displays the output of the program:

```
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.jst.jdt.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v20211116-1657\jre\bin
Boolean value: true
double value: 1.0
```

The bottom of the image shows the laptop keyboard, specifically the function keys (F1-F12) and other special keys like Esc, PgUp, PgDn, and Print Sc.

EXPLICIT CONVERSION

BOOLEAN TO CHARACTER

The screenshot shows the Eclipse IDE interface. The top part displays the code editor with a Java file named 'first.java'. The code defines a class 'first' with a main method that prints 'true' as a character ('T') using a ternary operator. The bottom part shows the 'Console' view with the output of the program: 'Boolean value: true' and 'Char value: T'.

```
first - first/src/first.java - Eclipse IDE
File Refactor Navigate Search Project Run Window Help
ConnectionDemo.java servlet.java apache-tomcat-9.0.71 at localhost first.java
Library [JavaSE-16]
package
va
public class first {
    public static void main(String[] args) {
        boolean b = true;
        char c = b ? 'T' : 'F'; // ternary operator to convert boolean to char
        System.out.println("Boolean value: " + b);
        System.out.println("Char value: " + c);
    }
}

Problems Javadoc Declaration Console
<terminated> first [Java Application] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.1.v
Boolean value: true
Char value: T

ASUS VivoBook
```

EXPLICIT CONVERSION

	byte	short	int	long	float	double	char	boolean
byte	-	E	E	E	E	E	E	E
short	I	-	E	E	E	E	E	E
int	I	I	-	E	E	E	E	E
long	I	I	I	-	E	E	E	E
float	I	I	I	I	-	E	E	E
double	I	I	I	I	I	-	E	E
char	E	E	E	E	E	E	-	E
bool	E	E	E	E	E	E	E	-

In this chart, "I" represents implicit conversion, "E" represents explicit conversion, and "-" represents no conversion needed.

Implicit conversion means that the conversion is done automatically by the compiler without any extra code needed. For example, when adding an int and a double, the int is implicitly converted to a double before the addition.

Explicit conversion means that the conversion requires an explicit cast operator to be used in the code. For example, when converting a double to an int, an explicit cast operator is needed to specify that the decimal part should be truncated.

Note that some conversions may result in a loss of precision or data, so it's important to be aware of the potential implications when performing type conversions.