

# RAGHAVENDRA KOWTAL

P:+918867003588|[github.com/Raghvkowtal](https://github.com/Raghvkowtal)|[raghvkowtal.github.io/Personal-Portfolio/](https://raghvkowtal.github.io/)|[linkedin.com/in/raghavendra-kowtal-583b1921a/](https://linkedin.com/in/raghavendra-kowtal-583b1921a/)

## EDUCATION

---

### SDM COLLEGE OF ENGINEERING AND TECHNOLOGY

Bachelor of Engineering

Dharwad, KA

Major in Electronics and Communications; Cumulative GPA: 7.23/10.0;

Aug 2019 - June 2023

Relevant Coursework: Web Development, Software Engineering; Operating Systems; Algorithms; Artificial Intelligence

### KARNATAKA SCIENCE COLLEGE

Pre-University Education

Dharwad, KA

2017 - 2019

## WORK EXPERIENCE

---

### Patil Electric Works Pvt Ltd

Armature Design Intern

Aug 2022 – Sep 2022

- Design of Armature according to the customer's requirements (Quality check, Assembly.)
- Manufacturer of armatures, dual speed motors & validate wiring harness in Hubballi.

### VIRAL FISSION

Student Ambassador

Jun 2021 – Sep 2021

- Promoted products digitally on Social-media platforms.
- Managed a team of 15-members, Supervision of product promotion on online platforms.

## UNIVERSITY PROJECTS

---

### REAL-TIME FOOD ORDERING SYSTEM

Aug 2022 – Mar 2023

- Designed and implemented a food ordering application using HTML, CSS, JavaScript, Node JS, My SQL.
- Rendered 3 interfaces namely Customer, Admin, Restaurant sides.
- Enabled users to order food using Intranet, where 3000+ people can order at same time. LINK- <https://tinyurl.com/2jx4tjc>

### REACT.JS PORTFOLIO WEBSITE

Jan 2023 – Apr 2023

- Created and rendered a portfolio design that can be used anyone.
- Technologies used: React.JS, HTML, CSS, JavaScript. LINK- <https://tinyurl.com/RECTP>

### REACT.SJ ONLINE EDUCATIONAL WEBSITE

Dec 2022 – Mar 2023

- Engineered and executed an educational website using React JS, JavaScript, HTML, CSS.
- Provided options to choose from 3 courses. LINK- <https://tinyurl.com/EDCWB>

### GESTURE TO SPEECH CONVERSION USING ML

Sep 2022 – Dec 2022

- Built a model using CNN Algorithm by providing 4000 Hand-gesture Data set with an Accuracy of 97%.
- Technologies used: ML, AI, Open-CV, Python, CNN, Tensorflow, Keras. LINK- <https://tinyurl.com/HGSRE>

### FACE RECOGNITION USING MACHINE LEARNING

Sep 2021 – Dec 2021

- Built a model using Haar Cascade Algorithm by providing 5000 face-images Data-set with an Accuracy of 95%.
- Trained the model using data-set of 5000 images.

## ACTIVITIES

---

### PUBLISHING

Published a paper entitled “Real-Time Intranet Based Food Ordering System” in [World Journal of Advanced Research and Reviews](https://wjarr.com/content/development-novel-real-time-intranet-based-food-ordering-system). Paper link-“<https://wjarr.com/content/development-novel-real-time-intranet-based-food-ordering-system>”.

### ADDITIONAL

---

**Programming Languages:** HTML, CSS, JavaScript, Python.

**Frameworks:** React JS, Material –UI, Bootstrap, Node.JS, Tailwind CSS.

**Developer Tools:** VS Code, PyCharm, IntelliJ.

**Certifications & Training:** Programming Concepts with ‘C’(ISCT-2019), MERN Stack Web Application Development (2023).

**Email-** raghavendrakowtal@gmail.com

NAME- RAGHAVENDRA KOWTAL

EMAIL-raghavendrakowtal@gmail.com

BATCH- A2 (2 – 4pm)

**TOPIC-**

## **PREMITIVE DATATYPES CONVERTIBILITY CHECK**

# CHAR CAN BE CONVETRED INTO BYTE BY EXPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'src' folder contains three files: 'Convertjava.java', 'Demo.java', and 'demo1.java'. The 'demo1.java' file is open in the editor, displaying the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         char a= 5;
6         byte b ;
7
8         b=(byte)a;
9         System.out.println(b);
10    }
11 }
12 //char can be converted into byte //
```

The code defines a class 'demo1' with a main method. It declares a character variable 'a' and initializes it to 5. It then declares a byte variable 'b' and initializes it to the value of 'a' using an explicit type cast. Finally, it prints the value of 'b' to the console. The output in the 'Console' view shows the number 5.

# CHAR CAN BE CONVETRED INTO SHORT BY EXPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         char a= 25;
6         short b ;
7
8         b=(short)a;
9         System.out.println(b);
10    }
11 }
12 //char can be converted into byte //
```

The code defines a class 'demo1' with a main method. It declares a character variable 'a' with the value 25 and a short variable 'b'. It then casts 'a' to a short and prints it. A comment at the end indicates that a character can be converted into a byte.

In the 'Console' view, the output is shown as:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 6:44:10 pm - 6:44:14 pm) [pid: 15140]
25
```

The system status bar at the bottom right shows the date and time as 20-07-2023 18:44.

# CHAR CAN BE CONVETRED INTO INT BY IMPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         char a= 'R';
6         int b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12 //char can be converted into byte //
```

The code defines a class 'demo1' with a main method. Inside the main method, a character variable 'a' is assigned the value 'R'. This value is then assigned to an integer variable 'b'. Finally, the value of 'b' is printed to the console using System.out.println. A comment at the end of the code explains that a character can be converted into a byte.

In the 'Console' view, the output is shown as:

```
82
```

The system output window also displays the following information:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 6:50:40 pm - 6:50:42 pm) [pid: 22116]
```

The taskbar at the bottom of the screen shows various application icons, and the system tray indicates the date and time as 20-07-2023, 18:51.

# CHAR CAN BE CONVENTRED INTO LONG BY IMPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         char a = 'A';
6         long b;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12 //char can be converted into byte //
```

The 'Console' tab shows the output of the program: '65'. The system tray at the bottom right indicates the date as 20-07-2023 and the time as 18:54.

# CHAR CAN BE CONVETRED INTO FLOAT BY EXPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         char a= 'M';
6         Float b ;
7
8         b=(float)a;
9         System.out.println(b);
10    }
11 }
12 //char can be converted into byte //
```

The code demonstrates that a character value ('M') can be explicitly cast to a float type (Float b). When run, the output is 77.0, as shown in the 'Console' tab.

System tray icons are visible at the bottom right, including a battery icon with a red circle.

# CHAR CAN BE CONVETRED INTO FLOAT BY EXPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         char a= 'F';
6         Double b;
7
8         b=(double)a;
9         System.out.println(b);
10    }
11 }
12 //char can be converted into byte //
```

The 'Console' tab shows the output of the program: 70.0. This indicates that the character 'F' was successfully converted to the float value 70.0.

# CHAR CANNOT BE CONVETRED INTO BOOLEAN BY EXPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'src' folder contains a 'test1' package with three files: 'Convertjava.java', 'Demo.java', and 'demol.java'. The 'demol.java' file is open in the editor, showing the following code:

```
1 package test1;
2
3 public class demol {
4     public static void main(String[] args) {
5         char a= 'K';
6         boolean b ;
7
8         b=(boolean)a; //ERROR
9         System.out.println(b);
10    }
11 }
12 //char can be converted into byte //
```

A red error marker is present on line 8, indicating a compilation error. The error message is displayed in the 'Console' view:

```
<terminated> demo1 [Java Application] C:\Users\raghav\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 7:00:23 pm - 7:00:26 pm) (pid: 20116)
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from char to boolean

        at test1/test1.demol.main(demol.java:8)
```

The status bar at the bottom right shows the date and time: 20-07-2023 19:00.

# BYTE CAN BE CONVETRED INTO SHORT BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         byte a = 10;
6         short b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
```

The output window (Console) shows the result of running the application:

```
10
```

The system tray at the bottom right shows the date and time as 20-07-2023, 19:52.

# BYTE CAN BE CONVETRED INTO INT BY IMPLICITE TYPE-CASTING

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

- Project Explorer:** Shows a project named "test1" with a "src" folder containing files: "Convert.java", "Demo.java", "demo1.java", and "module-info.java".
- Code Editor:** Displays the content of "demo1.java":

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         byte a= 56;
6         int b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
```

- Console:** Shows the output of the application run: "56".
- System Tray:** Shows icons for Search, Task View, File Explorer, Task List, and others.
- Bottom Bar:** Includes "Writable", "Smart Insert", and a timestamp "5 : 19 : 94".
- System Status:** Shows "ENG IN" and a date/time stamp "20-07-2023 19:57".

# BYTE CAN BE CONVETRED INTO LONG BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'src' folder contains a package 'test1' with three files: 'Convert.java', 'Demo.java', and 'demo1.java'. The 'demo1.java' file is open in the editor, displaying the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         byte a= 99;
6         long b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12 |
```

The 'Console' tab at the bottom shows the output of the program: '99'. The status bar at the bottom right indicates the date and time as '20-07-2023 19:58'.

# BYTE CAN BE CONVETRED INTO FLOAT BY IMPLICITE TYPE-CASTING

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

- Project Explorer:** Shows a project named "test1" with a package "test1" containing files "ConvertJava.java", "Demo.java", "demo1.java", and "module-info.java".
- Code Editor:** Displays the content of "demo1.java":

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         byte a= 75;
6         float b ;
7
8         b=(float)a;
9         System.out.println(b);
10    }
11 }
```

- Console:** Shows the output of the application run:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 8:09:45 pm – 8:09:48 pm) [pid: 13504]
75.0
```

- System Tray:** Shows icons for search, battery, signal strength, and date/time (20-07-2023).

# BYTE CAN BE CONVETRED INTO DOUBLE BY IMPLICITE TYPE-CASTING

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

- Project Explorer:** Displays a Java project named "test1" with a package "test1" containing a class "demo1". The file "demo1.java" is open, showing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         byte a= 122;
6         double b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
```
- Console:** Shows the output of the program execution:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 8:23:06 pm - 8:23:07 pm) [pid: 964]
122.0
```
- System Tray:** Shows various system icons including a battery icon.
- Taskbar:** Shows the Windows taskbar with icons for File Explorer, Search, Task View, Task Manager, and other applications like Microsoft Edge and Google Chrome.

# BYTE CANNOT BE CONVETRED INTO BOOLEAN BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         byte a= 122;
6         boolean b ;
7
8         b=(boolean)a;
9         System.out.println(b);
10    }
11 }
```

A red circle with a 'C' indicates a compilation error on line 8. The 'Console' tab shows the following output:

```
<terminated> demo1 [Java Application] C:\Users\raghav\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 8:25:18 pm – 8:25:23 pm) [pid: 12476]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from byte to boolean
        at test1/test1.demo1.main(demo1.java:8)
```

The system tray at the bottom right shows the date and time as 20-07-2023 20:27.

# SHORT CANNOT BE CONVENTRED INTO BYTE BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         short a= 1200;
6         byte b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12
```

In line 8, there is a compilation error: `b=a;`. The error message in the 'Console' view is:

```
<terminated> demo1 [Java Application] C:\Users\ragha\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 8:42:50 pm – 8:43:21 pm) [pid: 7220]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Type mismatch: cannot convert from short to byte

at test1/test1.demo1.main(demo1.java:8)
```

The system tray at the bottom of the screen shows various icons, including a battery icon indicating low power.

# SHORT CAN BE CONVETRED INTO INT BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         short a= 31000;
6         int b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab at the bottom shows the output of the program: '31000'. This demonstrates that the value of the short variable 'a' is implicitly converted to an int when assigned to the int variable 'b'.

# SHORT CAN BE CONVETRED INTO LONG BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         short a= 31000;
6         float b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12
```

The code attempts to assign the value of a short variable 'a' (31000) to a float variable 'b'. This assignment is implicitly cast by Java, resulting in the value 31000.0 being printed to the console.

In the Eclipse interface, the 'Console' view shows the output:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 8:54:18 pm - 8:54:19 pm) [pid: 17864]
31000.0
```

The system tray at the bottom right of the screen shows the date and time as 20-07-2023 20:54.

# SHORT CAN BE CONVETRED INTO FLOAT BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         short a= 28000;
6         float b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12
```

The code defines a class 'demo1' with a main method. It declares a short variable 'a' with the value 28000 and a float variable 'b'. It then assigns the value of 'a' to 'b' and prints 'b' to the console. The output in the 'Console' view shows '28000.0', indicating that the short value was implicitly converted to a float before being printed.

# SHORT CANNOT BE CONVERTED INTO DOUBLE BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         short a= 28000;
6         double b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12
```

The code attempts to assign the value of a short variable 'a' (which is 28000) to a double variable 'b'. This assignment is implicitly cast by the Java compiler, but it results in a loss of precision because the value 28000 cannot be exactly represented as a double. The output of the program, as shown in the 'Console' view, is '28000.0', indicating that the value has been rounded.

# SHORT CANNOT BE CONVETRED INTO BOOLEAN BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         short a= 22444;
6         boolean b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
```

In the 'Console' view, an error message is displayed:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 8:30:25 pm - 8:30:29 pm) [pid: 21092]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Type mismatch: cannot convert from short to boolean
at test1/test1.demo1.main(demo1.java:8)
```

The error message indicates that the compiler is unable to convert the type of variable 'a' (short) to 'b' (boolean) at line 8.

# INT CAN BE CONVETRED INTO CHAR BY EXPLICTE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         int a= 2854000;
6         char b ;
7
8         b=(char)a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab shows the output of the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav\p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 8:57:26 pm - 8:57:27 pm) [pid: 14860]
28000.0
```

The system tray at the bottom right shows the date and time as 20-07-2023 21:02.

# INT CAN BE CONVETRED INTO BYTE BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         int a= 2854000;
6         byte b ;
7
8         b=(byte)a; //ERROR
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab shows the output of the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 9:07:40 pm - 9:07:41 pm) [pid: 17188]
112
```

The status bar at the bottom right indicates the date and time: 20-07-2023, 21:07.

# INT CAN BE CONVETRED INTO SHORT BY EXPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         int a= 2854000;
6         short b ;
7
8         b=(short)a;
9         System.out.println(b);
10    }
11 }
12
```

The code demonstrates explicit type casting by assigning the value of 'a' (an int) to 'b' (a short) using the cast operator '(short)a'. The output of the program is shown in the 'Console' tab, where the value '112' is printed.

# INT CAN BE CONVETRED INTO LONG BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         int a= 254000;
6         long b ;
7
8         b=(long)a;
9         System.out.println(b);
10    }
11 }
12
```

The code defines a class 'demo1' with a main method. It declares an integer variable 'a' with the value 254000 and a long variable 'b'. It then casts 'a' to a long and prints it to the console. The output in the 'Console' view shows the value 254000.

# INT CAN BE CONVETRED INTO FLOAT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         int a= 24000;
6         float b ;
7
8         b=(float)a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab at the bottom shows the output of the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav\p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 9:33:47 pm – 9:33:48 pm) [pid: 17384]
24000.0
```

The system tray at the bottom right shows the date and time as 20-07-2023 21:33.

# INT CAN BE CONVETRED INTO DOUBLE BY IMPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         int a= 24000;
6         double b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab at the bottom shows the output of the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 9:35:09 pm - 9:35:09 pm) [pid: 12220]
24000.0
```

The system tray icons indicate the date as 20-07-2023 and the time as 21:35.

# INT CAN BE CONVETRED INTO BOOLEAN BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         int a= 23400;
6         boolean b ;
7
8         b=(boolean)a;//ERROR
9         System.out.println(b);
10    }
11 }
12
```

A red squiggly underline is under the assignment statement `b=(boolean)a;`, indicating a compilation error. The error message is displayed in the 'Console' view:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 9:37:14 pm – 9:37:26 pm) [pid: 12572]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from int to boolean

        at test1/test1.demo1.main(demo1.java:8)
```

The status bar at the bottom right shows the date and time: 20-07-2023, 21:37.

# LONG CAN BE CONVETRED INTO CHAR BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         long a= 23434500;
6         char b ;
7
8         b=(char)a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab shows the output of the program: '锄'. The status bar at the bottom right indicates the date and time as '20-07-2023 21:46'.

# LONG CAN BE CONVETRED INTO BYTE BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         long a= 23434500;
6         byte b ;
7         |
8         b=(byte)a;
9         System.out.println(b);
10    }
11 }
12
```

The code demonstrates explicit type casting from a long variable 'a' to a byte variable 'b'. The output of the program is '4', as shown in the 'Console' tab.

# LONG CAN BE CONVETRED INTO SHORT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         long a= 23434500;
6         short b ;
7
8         b=(short)a;
9         System.out.println(b);
10    }
11 }
12
```

The line `b=(short)a;` is highlighted with a red bracket. The output window shows the result of the run:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 9:48:57 pm - 9:48:57 pm) (pid: 12356)
-27388
```

The status bar at the bottom right indicates the date and time as 20-07-2023 and 21:49.

# LONG CAN BE CONVENTRED INTO INT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         long a= 23434500;
6         int b ;
7
8         b=(int)a;
9         System.out.println(b);
10    }
11 }
12
```

The code defines a class 'demo1' with a main method. It declares a long variable 'a' with the value 23434500 and an int variable 'b'. It then casts 'a' to an int and prints it to the console. The output in the 'Console' view shows the value 23434500.

# LONG CAN BE CONVETRED INTO FLOAT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         long a= 23434500;
6         float b ;
7
8         b=(float)a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab shows the output of the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 9:51:19 pm - 9:51:22 pm) [pid: 21340]
2.34345E7
```

The system tray at the bottom right shows the date and time as 20-07-2023 21:51.

# LONG CAN BE CONVETRED INTO DOUBLE BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         long a = 23434500;
6         double b;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12
```

The variable 'a' is explicitly cast to 'b' using the assignment operator 'a'. In the 'Console' view, the output is displayed as '2.34345E7', indicating that the long value was converted to a double before being printed.

# LONG CAN BE CONVETRED INTO BOOLEAN BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         long a= 23434500;
6         boolean b ;
7
8         b=(boolean) a;
9         System.out.println(b);
10    }
11 }
12
```

The line `b=(boolean) a;` is highlighted with a red squiggly underline, indicating a compilation error. The 'Console' view shows the following output:

```
<terminated> demo1 [Java Application] C:\Users\ragha\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 9:55:27 pm - 9:55:32 pm) [pid: 21792]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from long to boolean

        at test1/test1.demo1.main(demo1.java:8)
```

The system tray at the bottom right shows the date and time as 20-07-2023 21:55.

# FLOAT CAN BE CONVETRED INTO CHAR BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         float a= 23432;
6         char b ;
7
8         b=(char)a;
9         System.out.println(b);
10    }
11 }
12
```

The code demonstrates explicit type casting by converting the float variable 'a' to a character variable 'b' using the cast operator '(char)a'. The output is displayed in the 'Console' tab, which shows the character '2'.

# FLOAT CAN BE CONVETRED INTO BYTE BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         float a= 23432;
6         byte b ;
7
8         b=(byte)a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab shows the output of the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:03:57 pm - 10:03:59 pm) [pid: 14256]
-120
```

The status bar at the bottom right indicates the date and time: 20-07-2023 22:04.

# FLOAT CAN BE CONVETRED INTO SHORT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         float a= 23432;
6         short b ;
7
8         b=(short)a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab at the bottom shows the output of the program: '23432'. This demonstrates that the float value 23432 is explicitly cast to a short type before being printed.

# FLOAT CAN BE CONVETRED INTO INT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         float a= 23443.32;
6         int b ;
7
8         b=(int)a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab shows the output of the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:06:18 pm - 10:06:19 pm) [pid: 25140]
2344332
```

The system tray at the bottom right shows the date and time as 20-07-2023 22:06.

# FLOAT CAN BE CONVETRED INTO LONG BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         float a= 23978;
6         long b ;
7
8         b=(long)a;
9         System.out.println(b);
10    }
11 }
12
```

The code defines a class 'demo1' with a main method. It declares a float variable 'a' with the value 23978 and a long variable 'b'. It then casts 'a' to a long and prints its value. The output is shown in the 'Console' tab:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:07:18 pm - 10:07:20 pm) [pid: 10928]
23978
```

The console output shows the value 23978, confirming the successful conversion.

# FLOAT CAN BE CONVERTED INTO DOUBLE BY IMPLICIT TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         float a= 23978;
6         double b ;
7
8         b=a;
9         System.out.println(b);
10    }
11 }
12
```

The variable 'a' is of type float, and 'b' is of type double. In the assignment statement 'b=a;', there is an implicit type cast from float to double. The output of the program, shown in the 'Console' tab, is '23978.0', demonstrating that the float value was converted to a double before being printed.

# FLOAT CAN BE CONVETRED INTO BOOLEAN BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface. In the top-left corner, the title bar reads "eclipse-workspace - test1/src/test1/demo1.java - Eclipse IDE". The menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, and Help. The Package Explorer view on the left shows a project structure with a package named "test1" containing a class "demo1" with a main method. The code editor window displays the following Java code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         float a= 23978;
6         boolean b ;
7
8         b=(boolean)a; //ERROR
9         System.out.println(b);
10    }
11 }
12
```

The line "b=(boolean)a;" is highlighted in pink, indicating a compilation error. The status bar at the bottom right shows the date and time as "20-07-2023 22:09". Below the code editor is a terminal window titled "Console" which outputs the following error message:

```
<terminated> demo1 [Java Application] C:\Users\ragha\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:09:04 pm - 10:09:07 pm) [pid: 21368]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from float to boolean

        at test1/test1.demo1.main(demo1.java:8)
```

# DOUBLE CAN BE CONVETRED INTO CHAR BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         double a= 3423978;
6         char b ;
7
8         b=(char)a;
9         System.out.println(b);
10    }
11 }
12
```

The code demonstrates explicit type casting of a double variable 'a' to a char variable 'b' using the cast operator '(char)a'. The output is displayed in the 'Console' tab, showing the character '8'.

# DOUBLE CAN BE CONVETRED INTO BYTE BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         double a= 3423978;
6         byte b ;
7
8         b=(byte) a;
9         System.out.println(b);
10    }
11 }
12
```

The line `b=(byte) a;` is highlighted with a red bracket. The output window shows the result of the run:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:14:47 pm - 10:14:49 pm) [pid: 3240]
-22
```

The status bar at the bottom right indicates the date and time as 20-07-2023 22:14.

# DOUBLE CAN BE CONVETRED INTO SHORT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         double a= 3423978;
6         short b ;
7
8         b=(short)a;
9         System.out.println(b);
10    }
11 }
12
```

The output window shows the result of running the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:15:29 pm - 10:15:31 pm) [pid: 17076]
16106
```

The system status bar at the bottom right indicates the date and time as 20-07-2023 22:15.

# DOUBLE CAN BE CONVETRED INTO INT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         double a= 3423978;
6         int b ;
7
8         b=(int) a;
9         System.out.println(b);
10    }
11 }
12
```

The code uses explicit type casting ((int)) to convert the double value 'a' to an integer 'b'. The output of the program is displayed in the 'Console' tab, showing the value '3423978'.

# DOUBLE CAN BE CONVETRED INTO LONG BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         double a= 3423978;
6         long b ;
7
8         b=(long)a;
9         System.out.println(b);
10    }
11 }
12
```

The code defines a class 'demo1' with a main method. It declares a double variable 'a' and a long variable 'b'. It then casts 'a' to a long and prints it to the console. The output in the 'Console' view shows the value 3423978.

# DOUBLE CAN BE CONVETRED INTO FLOAT BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         double a= 3423978;
6         float b ;
7
8         b=(float)a;
9         System.out.println(b);
10    }
11 }
12
```

The 'Console' tab shows the output of the program:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk\hotspot\jre\full\win32\x86_64\17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:17:59 pm - 10:18:01 pm) [pid: 24028]
3423978.0
```

The system status bar at the bottom right indicates the date as 20-07-2023 and the time as 22:18.

# DOUBLE CAN BE CONVETRED INTO BOOLEAN BY EXPLICITE TYPE-CASTING

The screenshot shows the Eclipse IDE interface. In the top-left corner, there's a title bar for "eclipse-workspace - test1/src/test1/demo1.java - Eclipse IDE". Below it is a menu bar with File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help. On the left, the Package Explorer shows a project structure with Application, JRE System Library [JavaSE-17], and a src folder containing test1 (with Convertjava, Demo.java, demo1.java) and module-info.java. The main editor area displays the following Java code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         double a= 3423978;
6         boolean b ;
7
8         b=(boolean)a; //RECOR
9         System.out.println(b);
10    }
11 }
12
```

In line 8, there is a syntax error indicated by a red squiggly underline under the assignment operator "(boolean)". The code is run in the terminal window below, which outputs the following error message:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:18:55 pm - 10:18:58 pm) [pid: 18460]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from double to boolean

        at test1/test1.demo1.main(demo1.java:8)
```

The terminal window also shows the system status bar at the bottom with "Writable", "Smart Insert", "8:31:147", "ENG IN", "22:18", and the date "20-07-2023".

# BOOLEAN CANNOT BE CONVETRED INTO CHAR

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         boolean a= true;
6         char b ;
7
8         b=(char)a; //REEDOR
9         System.out.println(b);
10    }
11 }
12
```

A red squiggly underline is under the assignment statement `b=(char)a;`, indicating a compilation error. The error message is displayed in the 'Console' view:

```
<terminated> demo1 [Java Application] C:\Users\ragha\p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:21:18 pm - 10:21:21 pm) [pid: 20352]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from boolean to char

        at test1/test1.demo1.main(demo1.java:8)
```

The status bar at the bottom right shows the date and time: 20-07-2023 22:25.

# BOOLEAN CANNOT BE CONVETRED INTO BYTE

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         boolean a = true;
6         byte b ;
7
8         b=(byte)a; //REEDOR
9         System.out.println(b);
10    }
11 }
12
```

A red squiggly underline is under the assignment statement `b=(byte)a;`, indicating a compilation error. The error message is displayed in the 'Console' view:

```
<terminated> demo1 [Java Application] C:\Users\raghav.p2\pool\plugins\org.eclipse.jdt.core\openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:26:29 pm - 10:26:32 pm) [pid: 16424]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from boolean to byte

        at test1/test1.demo1.main(demo1.java:8)
```

The status bar at the bottom right shows the date and time as 20-07-2023 22:26.

# BOOLEAN CANNOT BE CONVETRED INTO SHORT

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         boolean a= true;
6         short b ;
7
8         b=(short)a; //RECOR
9         System.out.println(b);
10    }
11 }
12
```

The line `b=(short)a;` is highlighted with a red underline, indicating a compilation error. The error message in the 'Console' view is:

```
<terminated> demo1 [Java Application] C:\Users\vaghav\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:27:27 pm - 10:27:29 pm) [pid: 17008]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from boolean to short

        at test1/test1.demo1.main(demo1.java:8)
```

The status bar at the bottom right shows the date and time as 20-07-2023 22:27.

# BOOLEAN CANNOT BE CONVETRED INTO INT

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         boolean a= true;
6         int b ;
7
8         b=(int)a; //REEDOR
9         System.out.println(b);
10    }
11 }
12
```

A red error marker is present on line 8, indicating a compilation error. The error message in the 'Console' view is:

```
<terminated> demo1 [Java Application] C:\Users\vagh\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:28:17 pm - 10:28:18 pm) [pid: 12560]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Cannot cast from boolean to int

at test1/test1.demo1.main(demo1.java:8)
```

The system tray at the bottom right shows the date and time as 20-07-2023 22:28.

# BOOLEAN CANNOT BE CONVETRED INTO LONG

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         boolean a= true;
6         long b ;
7
8         b=(long)a; //REEDOR
9         System.out.println(b);
10    }
11 }
12
```

The line `b=(long)a;` is highlighted with a red underline, indicating a compilation error. The error message in the 'Console' view is:

```
<terminated> demo1 [Java Application] C:\Users\vagh\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:29:02 pm - 10:29:04 pm) [pid: 5884]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Cannot cast from boolean to long

at test1/test1.demo1.main(demo1.java:8)
```

The status bar at the bottom right shows the date and time as 20-07-2023 22:29.

# BOOLEAN CANNOT BE CONVETRED INTO FLOAT

The screenshot shows the Eclipse IDE interface with a Java project named "test1". The "demo1.java" file is open in the editor, containing the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         boolean a= true;
6         float| b ;
7
8         b=(float)a; //REEDOR
9         System.out.println(b);
10    }
11 }
12
```

A red squiggly underline is under the assignment statement `b=(float)a;`, indicating a compilation error. The error message in the "Console" view is:

```
<terminated> demo1 [Java Application] C:\Users\vaghav\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:29:42 pm - 10:29:44 pm) [pid: 2924]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from boolean to float

        at test1/test1.demo1.main(demo1.java:8)
```

The status bar at the bottom right shows the date and time as "20-07-2023 22:29".

# BOOLEAN CANNOT BE CONVETRED INTO DOUBLE

The screenshot shows the Eclipse IDE interface with a Java project named 'test1'. The 'demo1.java' file contains the following code:

```
1 package test1;
2
3 public class demo1 {
4     public static void main(String[] args) {
5         boolean a= true;
6         double b;
7
8         b=(double)a; //REEROR
9         System.out.println(b);
10    }
11 }
12
```

The line `b=(double)a;` is highlighted with a red underline, indicating a compilation error. The error message in the 'Console' view is:

```
<terminated> demo1 [Java Application] C:\Users\vagh\p2\pool\plugins\org.eclipse.jdt.openjdk.hotspot.jre.full.win32.x86_64.17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023, 10:30:32 pm - 10:30:33 pm) [pid: 13656]
Exception in thread "main" java.lang.Error: Unresolved compilation problem:
    Cannot cast from boolean to double

        at test1/test1.demo1.main(demo1.java:8)
```

The status bar at the bottom right shows the date and time as 20-07-2023 22:30.

## 8 X 8 TABLE OF REMITIVE DATATYPES

	CHAR	BYTE	SHORT	INT	LONG	FLOAT	DOUBLE	BOOLEAN
CHAR	NCR	EC	EC	IC	IC	EC	EC	NC
BYTE	EC	NCR	IC	IC	IC	EC	IC	NC
SHORT	EC	IC	NCR	IC	IC	IC	IC	NC
INT	EC	EC	EC	NCR	EC	EC	EC	NC
LONG	EC	EC	EC	EC	NCR	EC	IC	NC
FLOAT	EC	EC	EC	EC	EC	NCR	IC	NC
DOUBLE	EC	EC	EC	EC	EC	EC	NCR	NC
BOOLEAN	NO	NO	NO	NO	NO	NO	NO	NCR

**NOTE :-**

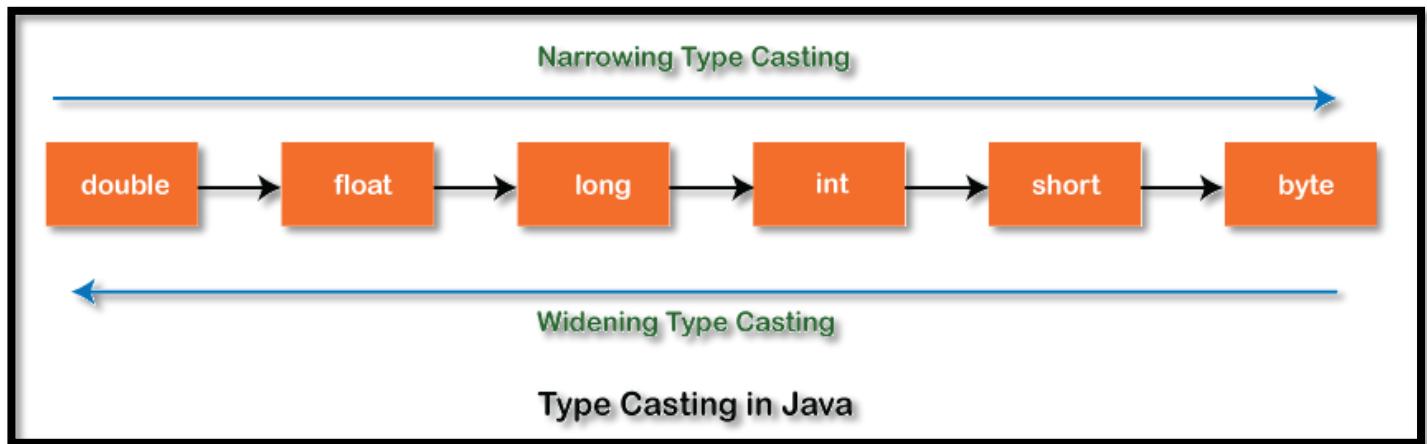
**NCR- NO CONVENTION REQUIRED**

**EC- EXPLICIT CONVERSION**

**IC- IMPLICIT CONVERSION**

**NC – NOT CONVERTABLE**

# TYPE CONVERSION GRAPH IN JAVA



# TODAY'S PROGRAMME

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Demo.java
3 import java.util.Scanner;
4
5 public class Demo {
6
7     public static void main(String[] args) {
8         Scanner scan = new Scanner(System.in);
9         System.out.println("Enter the name of student");
10        String a = scan.nextLine();
11        System.out.println("Name of the student is "+a);
12        System.out.println("What is the age of student");
13        int b = scan.nextInt();
14        System.out.println("Age of the student is "+ b);
15        System.out.println("What is the CGPA of student");
16        Float c = scan.nextFloat();
17        System.out.println("CGPA of the student is "+ c);
18        System.out.println("Is the student married ? say True or False");
19        Boolean d = scan.nextBoolean();
20        System.out.println("value stored in d is "+ d);
21        System.out.println("What is the height of student");
22        Double e = scan.nextDouble();
23        System.out.println("height of the student is "+ e);
24        System.out.println("What is the weight of student");
25        Double f = scan.nextDouble();
26        System.out.println("Weight of the student is "+ f);
27        System.out.println("Enter the BRANCH of student");
28        scan.nextLine();
29        String g = scan.nextLine();
30        System.out.println("Branch of the student is "+ g);
31    }
32 }
33
34 }
```

Console <terminated> Demo (1) [Java Application] C:\Users\raghav\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023 10:46:35 pm - 10:47:18 pm) [pid: 14860]

```
Enter the name of student
```

## OUTPUT:

```
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Demo.java
25 Double f = scan.nextDouble();
```

Console <terminated> Demo (1) [Java Application] C:\Users\raghav\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86\_64\_17.0.7.v20230425-1502\jre\bin\javaw.exe (20-Jul-2023 10:46:35 pm - 10:47:18 pm) [pid: 14860]

```
Enter the name of student
RAGHAVENDRA KOWTAL
Name of the student is RAGHAVENDRA KOWTAL
What is the age of student
23
Age of the student is 23
What is the CGPA of student
7.23
CGPA of the student is 7.23
Is the student married ? say True or False
FALSE
value stored in d is false
What is the height of student
5.8
height of the student is 5.8
What is the weight of student
70
Weight of the student is 70.0
Enter the BRANCH of student
ELECTRONICS AND COMMUNICATION
Branch of the student is ELECTRONICS AND COMMUNICATION
```