# Constant in Java - Example

## Points to Remember:

- Write the identifier name in capital letters that we want to declare as constant. For example, **MAX=12**.
- If we use the **private** access-specifier before the constant name, the value of the constant cannot be changed in that particular class.
- If we use the **public** access-specifier before the constant name, the value of the constant can be changed in the program.

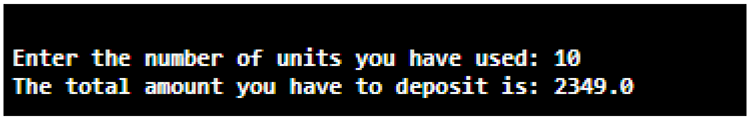
Let's see some examples in which we have used constants.

## Example 1: Declaring Constant as Private

**ConstantExample1.java**

```
import java.util.Scanner;
public class ConstantExample1
{
//declaring constant
private static final double PRICE=234.90;
public static void main(String[] args)
{
int unit;
double total_bill;
System.out.print("Enter the number of units you have used: ");
Scanner sc=new Scanner(System.in);
unit=sc.nextInt();
total_bill=PRICE*unit;
System.out.println("The total amount you have to deposit is: "+total_bill);
}
}
```

**Output:**



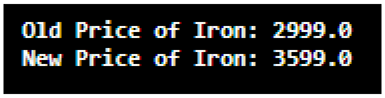
```
Enter the number of units you have used: 10
The total amount you have to deposit is: 2349.0
```

## Example 2:

**ConstantExample2.java**

```
public class ConstantExample2
{
private static final double PRICE=2999;
public static void main(String[] args)
{
System.out.println("Old Price of Iron: "+PRICE);
ConstantExample obj = new ConstantExample();
obj.showPrice();
}
}
class ConstantExample
{
private static final double PRICE=3599;
void showPrice()
{
System.out.print("New Price of Iron: "+PRICE);
}
}
```

**Output:**



```
Old Price of Iron: 2999.0
New Price of Iron: 3599.0
```

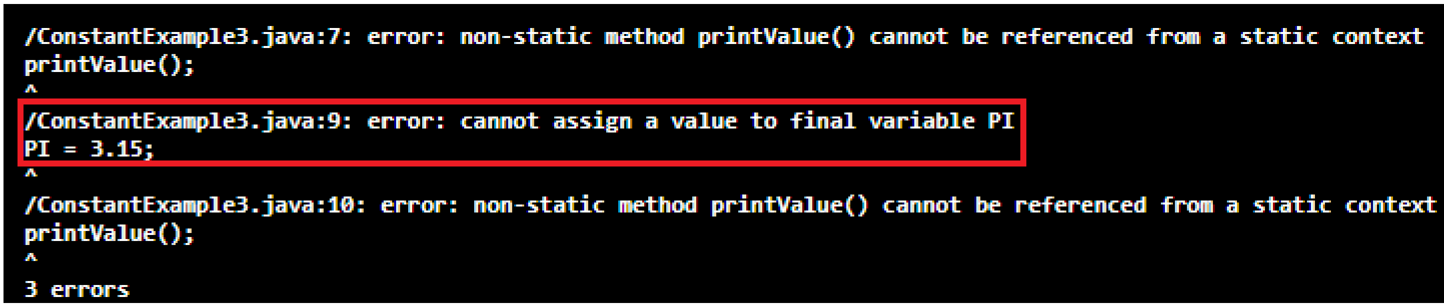
## Example 3: Declaring Constant as Public

In the following example, we have declared constant PI as public. Inside the main() method, we have assigned 3.15 in the constant PI. After that, we have invoked the printValue() method. When we execute the program, it shows an error cannot assign a **value to the final variable PI**.

**ConstantExample3.java**

```
public class ConstantExample3
{
//declaring PI as constant
public static final double PI= 3.14;
public static void main(String[] args)
{
printValue();
//trying to assign 3.15 in the constant PI
PI = 3.15;
printValue();
}
void printValue()
{
System.out.print("The value of PI cannot be changed to " + PI);
}
}
```

**Output:**



```
/ConstantExample3.java:7: error: non-static method printValue() cannot be referenced from a static context
printValue();
^
/ConstantExample3.java:9: error: cannot assign a value to final variable PI
PI = 3.15;
^
/ConstantExample3.java:10: error: non-static method printValue() cannot be referenced from a static context
printValue();
^
3 errors
```

## Using Enumeration (Enum) as Constant

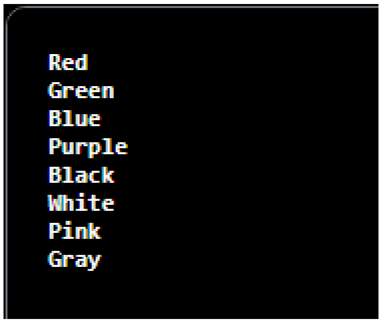
- It is the same as the final variables.
- It is a list of constants.
- Java provides the enum keyword to define the enumeration.

- It defines a class type by making enumeration in the class that may contain instance variables, methods, and constructors.

## Example of Enumeration

```
public class EnumExample
{
//defining the enum
public enum Color {Red, Green, Blue, Purple, Black, White, Pink, Gray}
public static void main(String[] args)
{
//traversing the enum
for (Color c : Color.values())
System.out.println(c);
}
}
```

**Output:**



```
Red
Green
Blue
Purple
Black
White
Pink
Gray
```