```java
package com.pack1 ;

public class ClassA
{
    int c;
    static int e;
    ClassA ()
    {
        System.out.println(++c);
        System.out.println(++e);
    }
    public static void main(String []args)
    {
        int a=34;
            int b=21;
            new ClassA().c-=a++ + ++b ;
            int d=--a + --b + new ClassA().c--;
            e=a + +b + +new ClassA().c + d--;
            int f=-a + b-- + -new ClassA().c - d++;
            int sum= a+ b + new ClassA().c + d + e + f;
            System.out.println("sum="+ sum);
        }
}
```

```
1
1
1
2
1
3
1
113
1
114
sum=156
```

Type Casting: It is a process of converting one datatype into another datatype (Except **boolean**)

Type casting

- Implicit Typecasting — Smaller Datatype ⇨ Larger Datatype (Widening)
  1) Implicit typecasting will be automatically done by the compiler
  2) In Implicit typecasting there is NO chance of loss of information
- Explicit Typecasting — Larger Datatype ⇨ Smaller Datatype (Narrowing)
  1) Explicit typecasting will not be done by the compiler, it is the responsibility of the programmer
  2) In Explicit typecasting there **may be** a chance of loss of information.

Understanding Type Casting

- Converting one data type into another data type is called casting.
- In general there are two types of casting procedures.
  - ✓ Implicit Type Casting
  - ✓ Explicit Type Casting

Implicit Type Casting:

- Converting smaller data type to larger data types is called "Implicit Type Casting".
- It is also known as Widening or Casting-Upwards.
- There is no lose of information in this type casting.

byte -> short, int, long, float, double
short -> int, long, float, double
char -> int, long, float, double
int -> long, float, double
long -> float, double
float -> double

Explicit Type Casting

- Converting larger data type to smaller data types is called "Explicit Type Casting".
- It is also known as Narrowing or Casting-Downwards.
- There may be a chance of lose of information in this type casting.

<Destination DataType> <variableName>=(DataType) <SourceType>

- Ex: int i=90;
- byte b = (byte)i;

byte -> char
short -> byte, char
char -> byte, short
int -> byte, short, char
long -> byte, short, char, int
float -> byte, short, char, int, long
double -> byte, short, char, int, long, float

In casting what happens if source variable has value greater than the destination variable type range?

- We will not get any compile time error or runtime error, assignment will be performed by reducing its value in the range of destination variable type range.
- We can know the value by using the below formula

[minimumRange + (result - maximumRange - 1)]

```java
package com.pack1 ;

public class ClassA
{
    void meth1()
    {
        System.out.println("Performing implicit typecasting\n");

        byte b=10;
        int i=b;
        System.out.println("byte PDT b : "+b);
        System.out.println("int PDT i : "+i);

    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
    }
}
```

```
Performing implicit typecasting

byte PDT b : 10
int PDT i : 10
```

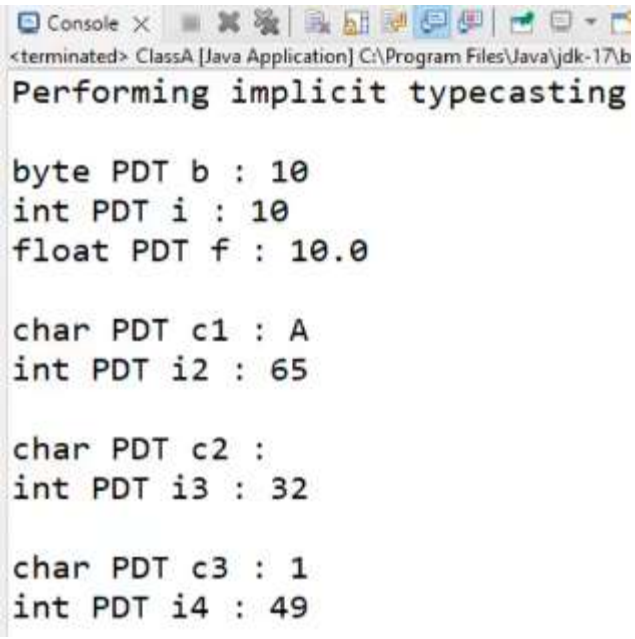
```java
package com.pack1 ;

public class ClassA
{
    void meth1()
    {
        System.out.println("Performing implicit typecasting\n");

        byte b=10;
        int i=b;
        float f=b;
        System.out.println("byte PDT b : "+b);
        System.out.println("int PDT i : "+i);
        System.out.println("float PDT f : "+f);

    }
    public static void main(String[] args)
    {
        ClassA aobj=new ClassA();
        aobj.meth1();
    }
}
```

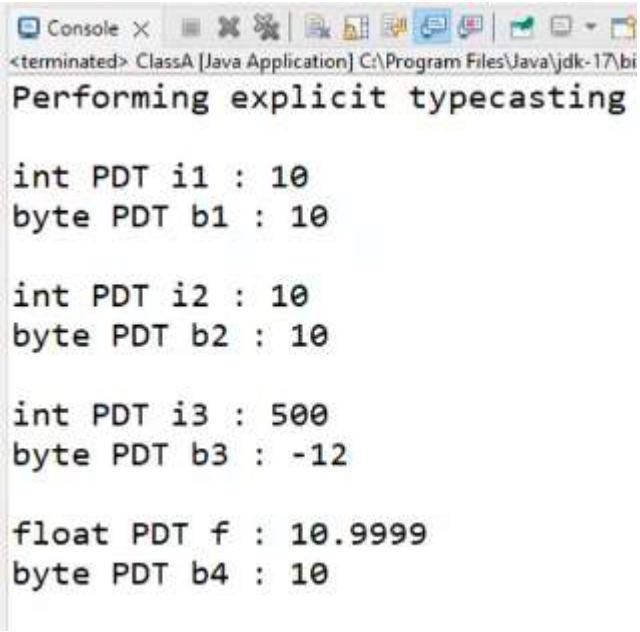
```
Performing implicit typecasting

byte PDT b : 10
int PDT i : 10
float PDT f : 10.0
```

```
 3 public class ClassA
 4 {
 5     void meth1()
 6     {
 7         System.out.println("Performing implicit typecasting\n");
 8
 9         byte b=10;
10         int i=b;
11         float f=b;
12         System.out.println("byte PDT b : "+b);
13         System.out.println("int PDT i : "+i);
14         System.out.println("float PDT f : "+f);
15
16         char c1='A';
17         int i2=c1;
18
19         char c2=' ';
20         int i3=c2;
21
22         char c3='1';
23         int i4=c3;
24

25         System.out.println("\nchar PDT c1 : "+c1);
26         System.out.println("int PDT i2 : "+i2);
27
28         System.out.println("\nchar PDT c2 : "+c2);
29         System.out.println("int PDT i3 : "+i3);
30
31         System.out.println("\nchar PDT c3 : "+c3);
32         System.out.println("int PDT i4 : "+i4);
33     }
34     void meth2()
35     {
36         System.out.println("Performing explicit typecasting");
37
38         int i1=10;
39         byte b1=(byte)i1;
40         System.out.println("\nint PDT i1 : "+i1);
41         System.out.println("byte PDT b1 : "+b1);
42
```

```
43          final int i2=10; // final variables are compile time constants
44          //i2=i2+200;// C.E
45          byte b2=i2;       // byte range is -128 to 127
46          System.out.println("\nint PDT i2 : "+i2);
47          System.out.println("byte PDT b2 : "+b2);
48
49          int i3=500;
50          byte b3=(byte)i3;
51          System.out.println("\nint PDT i3 : "+i3);
52          System.out.println("byte PDT b3 : "+b3);
53          /*
54           [minimumRange + (result - maximumRange - 1)]
55           -128+(500-127-1)===>-128+(500-128)===>-128+372===>244
56           -128+(244-127-1)===>-128+(244-128)===>-128+116===>-12
57           */
58
59          float f=10.9999f;
60          byte b4=(byte)f;
61          System.out.println("\nfloat PDT f : "+f);
62          System.out.println("byte PDT b4 : "+b4);
63      }
```

```
64      public static void main(String[] args)
65      {
66          ClassA aobj=new ClassA();
67          aobj.meth1();
68          aobj.meth2();
69      }
70  }
```


Console
<terminated> ClassA [Java Application] C:\Program Files\Java\jdk-17\b
Performing implicit typecasting
byte PDT b : 10
int PDT i : 10
float PDT f : 10.0
char PDT c1 : A
int PDT i2 : 65
char PDT c2 :
int PDT i3 : 32
char PDT c3 : 1
int PDT i4 : 49


Console
<terminated> ClassA [Java Application] C:\Program Files\Java\jdk-17\bi
Performing explicit typecasting
int PDT i1 : 10
byte PDT b1 : 10
int PDT i2 : 10
byte PDT b2 : 10
int PDT i3 : 500
byte PDT b3 : -12
float PDT f : 10.9999
byte PDT b4 : 10

```
3  public class ClassB
4  {
5      void Implicit()
6       {
7          byte b=10;
8          System.out.println("byte value is "+b);
9          short s=b;
10         System.out.println("short value is "+s);
11         s++;
12         int i=s++;
13         System.out.println("int value is "+i);
14         System.out.println("short value is "+s);
15         long l=i;
16         System.out.println("long value is :"+(--l));
17         float f=l;
18         System.out.println("The value of float is "+(f+b));
19         double d=(--f);
20         System.out.println("the value of double is "+d);
21         show();
22         if(!(d==f))
23         {
24             System.out.println("equal");
25         }
```

```
26         else
27         {
28             System.out.println("not equal");
29         }
30     }
31     public static void main(String[] args)
32     {
33         ClassB b=new ClassB();
34         b.Implicit();
35     }
36     static void show()
37     {
38         char c='A';
39         int a=++c;
40         System.out.println(a);
41         ClassB obj=new ClassB();
42         String s=obj.meth1();
43         System.out.println(s);
44     }
```

```
45      String meth1()
46      {
47          String S="Implicit casting is done by the compiler automatically";
48          return S;
49      }
50  }
```

```
13          System.out.println("int value is "+i
14          System.out.println("short value is "
15          long l=i;
16          System.out.println("long value is :"
17          float f=l;
18          System.out.println("The value of flo
19          double d=(--f);
20          System.out.println("the value of doul
21          show();
22          if(!(d==f))
23          {
24              System.out.println("equal");
25          }
26          else
27          {
28              System.out.println("not equal");
29          }
30      }
```

```
byte value is 10
short value is 10
int value is 11
short value is 12
long value is :10
The value of float is 20.0
the value of double is 9.0
66
Implicit casting is done by the compiler automatically
not equal
```

Assignment on Typecasting

Suppose when a customer deposits money, the bank calculates interest using floating-point , but the system primarily stores balances as long for efficiency.

Tasks:

1) Store a customer's initial balance as an int and convert it to long.

2) Calculate the interest (e.g., 4.5% of balance) and store it in a double.

3) Add interest to the balance

4) Print all values and verify type conversion.