```
int i1 = 5;                 // ok; default type for non-decimal is int
int i2 = 5.2;               // 5.2 is not an integer and does not implicitly cast

float f1 = 2;               // 2 will be implicitly casted to a float (lossless)
float f2 = 3.0f;            // "f" signifies value as float
float f2 = 3.0;             // 3.0 is a double, not a float (f smaller than d)
float f3 = 3.5;             // 3.5 is a double, not a float

double d1 = 3.5;            // ok; default type for decimal is double
double d2 = 2.0;            // ok
double d3 = 4;              // int can be implicitly casted to double (lossless)
double d4 = 3.5d;           // ok; d signifies value as double

i1 = (int) d1;              //explicit casting of a double into an int
System.out.println ("i = " + i1);     //prints i = 3

//i1 = 5.0 / 9.0;           //error because double can't be automatically   //
                            converted to int
i1 = 5 / 9;                 // division of integers yields integer
System.out.println ("i = " + i1);     // prints i = 0

f1 = (float) d1;            // explicitly casted
System.out.println ("f = " + f1);     // f = 3.5


f1 = 5 / 9;                 // integer division; casted to float
System.out.println ("f = " + f1);     // f = 0.0
f1 = 5.0/9.0;               // division results in a double, not a float
f1 = 5.0f / 9.0f;           // 0.5555556…
System.out.println ("f = " + f1);     // f = 0.5555556

d1 = 3.5 / 2.6;             // decimal division; returns 1.346153846153846
System.out.println ("d = " + d1);     // d = 1.346153846153846

d1 = (int) 3.5 / 2.6;       // 3.5 casted to int, then div; 1.1538461538461537
System.out.println ("d = " + d1);     // d = 1.1538461538461537

d1 = (int) (3.5) / 2.6; // 3.5 casted to int, then div; 1.1538461538461537
System.out.println ("d = " + d1);     // d = 1.1538461538461537

d1 = (int) (3.5 / 2.6); // 1.0
System.out.println ("d = " + d1);     // d = 1.0

d1 = int 3.5 / 2.6;         // improper casting syntax


d1 = (int) (3.5 / 2.6); // cast the entire expr *after* division; 1.0
System.out.println ("d = " + d1);     // d = 1.0

d1 = 3.5 / (int) 2.6;       // cast just 2.6 to int; 1.75
System.out.println ("d = " + d1);     // d = 1.75

d1 = (float) (int) (3.5 / 2.6);       // cast to int, then float; 1.0
System.out.println ("d = " + d1);     // d = 1.0

short smallValue = 45;   // short holds smaller integer values; 45
short s = 3.5;              // short cannot hold decimal values (ie. double)
```

```
smallValue = 234251434324324;  // the value is too large and overflows

int littleValue = smallValue; // int is larger than short; no loss; 45

smallValue = (short) littleValue;    // cast is valid as 45 fits in a short
System.out.println ("smallValue = " + smallValue);      // smallValue = 45
smallValue = (short) 234251434;      // truncates the number to 25770
System.out.println ("smallValue = " + smallValue);      // smallValue = 25770

int over = 1111111111111;      // integer overflow; number is too large to fit

float pay = 42234.45f;   // assigns 42234.45 to the float variable 'pay'
long bigValue = 45243224L;     // assigns 45253224 the long 'bigValue'
double amount = 345.45d;       // assigns 345.45 to the double 'amount'
```