Homework – Casting

1. Copy the following code into the main of a new class.
   - for each **assignment** statement write a note about what conversions take place &/or what rules are being used
   - for each **output** statement, write what the output is.
   - for each **error line**, take away the // to view the error message and describe why that line causes an error (in your own words)

Note: Some examples are done for you in red below

Some of the data types used here are ones that were outlined in the supplemental document `04_Input_Variable` that you were assigned as homework and you DEFINITELY read, right? 😉

---------- code -------------------------

```
int i1 = 5;
//ok; default type for non-decimal is int

//int i2 = 5.2;
//loss of information when assigning a double (5.2) to int (i2)

float f1 = 2;
//ok

float f2 = 3.0f;
//ok; f signifies to interpret as float

//float f2 = 3.0;
//loss of information when assigning a double (3.0) to float (f2)

//float f3 = 3.5;
//loss of information when assigning a double to a float

double d1 = 3.5;
//ok; default type for decimal is double

double d2 = 2.0;
//ok

double d3 = 4;
//ok

double d4 = 3.5d;
//ok; d signifies to interpret 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;
//ok; division of integers yields integer

System.out.println ("i = " + i1);
// prints i = 0

f1 = (float) d1;
//explicit casting of a double into a float

System.out.println ("f = " + f1);
//prints f = 3.5


f1 = 5 / 9;
// ok; conversion of integer (integer division) to float

System.out.println ("f = " + f1);
//prints f = 0.0

//f1 = 5.0/9.0;
//5.0/9.0 is real division and produces a double. A double cannot be
automatically converted to a float due to loss of data.

f1 = 5.0f / 9.0f;
//ok; dividing two floats

System.out.println ("f = " + f1);
// prints f = 0.555555555555555555555555555555555556…

d1 = 3.5 / 2.6;
//ok; dividing two doubles

System.out.println ("d = " + d1);
// prints d = 1.346153846153846…

d1 = (int) 3.5 / 2.6;
//ok; casting of 3.5 into an int and dividing 3 by 3.6 (follows BEDMAS)

System.out.println ("d = " + d1);
// prints d = 1.153846153846…


d1 = (int) (3.5) / 2.6;
//ok; casting of 3.5 into an int and dividing 3 by 3.6 (brackets are
redundant)

System.out.println ("d = " + d1);
// prints d = 1.153846153846…
```

```
d1 = (int) (3.5 / 2.6);
// casting final quotient (double) into an int and again explicitly
casting int into a double

System.out.println ("d = " + d1);
// prints d = 1.0

//d1 = int 3.5 / 2.6;
// no brackets around int. will result in a syntax error ?


d1 = (int) (3.5 / 2.6);
// casting final quotient (double) into an int and again explicitly
casting int into a double

System.out.println ("d = " + d1);
// prints d = 1.0

d1 = 3.5 / (int) 2.6;
// casting double (2.6) into an int and performing real division of
3.5/2.

System.out.println ("d = " + d1);
// prints d = 1.75

d1 = (float) (int) (3.5 / 2.6);
// casts the double result of real division into an int, then casts the
int into a float, and explicitly casting int into a double.

System.out.println ("d = " + d1);
// prints d = 1.0

short smallValue = 45;
//ok, 45 is enough info for a short.

//short s = 3.5;
// cannot automatically cast a double into a short, as there is a concern
for loss of information.

//smallValue = 234251434324324;
// 234251434324324 will lose information when casted into a short.

int littleValue = smallValue;
//ok, automatically casts a short into an int (more information)

smallValue = (short) littleValue;
// ok, explicitly casting a int into a short, then assigning a short to
an int (smallValue)

System.out.println ("smallValue = " + smallValue);
// prints smallValue = 45

smallValue = (short) 234251434;
// ok, explicitly casting an int into a short, then assiging a short to
an int (smallValue)

System.out.println ("smallValue = " + smallValue);
```

```
//prints smallValue = 25770

//int over = 1111111111111;
//error because integer number is too large

float pay = 42234.45f;
// ok; f signifies to interpret as float

long bigValue = 45243224L;
// ok; L signifies to interpret as long

double amount = 345.45d;
// ok; d signifies to interpret as double
```

2. **Average.java** Write a program that prompts the user for five grades and then displays the average of the grades. The grades are integers and they must be stored in variables of type `int`. Real division should be performed when calculating the average.

Exercises continue on next page

3. **Change_New.java** Create a program that prompts the user for an amount in dollar and then displays the minimum number of coins necessary to make the change. The change can be made up of toonies ($2), loonies ($1), quarters (25 cents), dimes (10 cents), nickels (5 cents), and pennies (1 cent). The program output should look similar to :

```
Enter the amount (in dollar): 5.34
The minimum number of coins is:
      Toonies: 2
      Loonies: 1
      Quarters: 1
      Dimes: 0
      Nickels: 1
      Pennies: 4
```

Hint: after you read in the dollar amount, convert it to the amount in cents (int) immediately.