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;          //error because...

float f1 = 2;           //ok
float f2 = 3.0f;        //ok; f signifies to interpret as float
//float f2 = 3.0;        //error since double is more complex than float
//float f3 = 3.5;        //error since double is more complex than float
                         //they both need explicit casting

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;              // 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, but does integer division so that is 0
System.out.println ("f = " + f1); // prints f = 0
//f1 = 5.0/9.0;          // does not use f suffix or cast explicitly to f
f1 = 5.0f / 9.0f;        // ok
System.out.println ("f = " + f1); // prints f = 0.5555556

d1 = 3.5 / 2.6; // ok
```

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

d1 = (int) 3.5 / 2.6; // ok, but assigns to d1 the int 3 divided by 2.6
System.out.println ("d = " + d1); // prints d = 1.1538461538461537

d1 = (int) (3.5) / 2.6; // ok, but assigns to d1 the int 3 divided by 2.6
System.out.println ("d = " + d1); // prints d = 1.1538461538461537

d1 = (int) (3.5 / 2.6); // ok, takes the real division of 3.5/2.6 and
floors it into an int
System.out.println ("d = " + d1); // prints d = 1.0

//d1 = int 3.5 / 2.6;   // error because there are no brackets around the
casting datatype


d1 = (int) (3.5 / 2.6); // ok, takes the real division of 3.5/2.6 and
floors it into an int
System.out.println ("d = " + d1); // prints d = 1.0

d1 = 3.5 / (int) 2.6; // ok, but assigns d1 3.5 divided by the int 2
System.out.println ("d = " + d1); // prints d = 1.75

d1 = (float) (int) (3.5 / 2.6); // does real division, floors it, then
casts it to float again explicitly
System.out.println ("d = " + d1); // prints d = 1.0

short smallValue = 45;           // ok
//short s = 3.5;                // error because 3.5 is a floating point
value
//smallValue = 234251434324324; //error because that exceeds the 2 byte
short limit

int littleValue = smallValue; // ok

smallValue = (short) littleValue; // ok
System.out.println ("smallValue = " + smallValue); // prints smallValue =
45
smallValue = (short) 234251434; // ok, but cuts off everything left of
two bytes
System.out.println ("smallValue = " + smallValue); // prints smallValue =
25770

//int over = 1111111111111; //error because that goes over the integer
limit (+/- 2^31-1)

float pay = 42234.45f; //ok, but not as accurate, stores 42234.449219
long bigValue = 45243224L; //ok
double amount = 345.45d; //ok
```

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.