Integer Division (Outsource: 3-52 – 3-53)

When you divide two integers, whether they are integer constants or integer variables, the result is an integer. In other words, any fractional part of the result is lost. For example, the result of **45 / 2** is **22** not **22.5**. Remember, an integer divided by an integer results in an integer. If you must divide two integers but want the result to be a whole number then you must use type casting. For example, the result of **(double) 45 / 2** is **22.5**. Notice that only the 45 has been type cast into a double. A double divided by an integer results in a double.

Modulus Division (Outsource: 3-54 – 3-55)

The modulus operator (**%**) is usually used with integers. It is also defined to work with float and double operands, though that use is quite rare. When you use the modulus division operator with two integers, the result is an integer with the value of the remainder after division takes place. For example, **22 % 4** is **2** because *22 divided by 4 is 5 with a remainder of 2*.

Examples of Modulus Division

Odd or Even Numbers

```
num % 2 = = 0;      /* Results in true when num is an integer value evenly divisible by  2.
                       In other words modulus division by 2 results in a remainder of  0 */

num % 2 = = 1;      /* Results in true when num is an integer value not evenly divisible
                       by 2. In other words modulus division by 2 results in a remainder of
                       1. NOTE: This does not work with negative numbers. */

num % 2 != 0;       /* This is a better way to test for odd numbers. This does work with
                       negative numbers */
```

How Many Weeks In numberOfDays

```
final int daysInAWeek = 7;  // The number of days in a week is constant
int numberOfDays = 23;

int numberOfWeeks = numberOfDays / daysInAWeek;  // this results in 3 (weeks)
int daysRemaining = numberOfDays % daysInAWeek;  // this results in 2 (days)
System.out.println(numberOfDays + " is the same as " + numberOfWeeks +
                   " weeks and " + daysRemaining + " days.");
```

LAB 05 - ASSIGNMENT

Lab 05A - 60 points

OBJECTIVE

WAP to input two integers from the keyboard. Divide the first number by the second number and display the results as a whole number.

FIELD SUMMARY

- **int dividend** – the number to be divided.
- **int divisor** – the number with which to divide.
- **int quotient** – the number of times divisor is contained in dividend.
- **int remainder** – the portion of the dividend that is not evenly divisible by the divisor.

METHOD SUMMARY

- **main** – instantiate an object of your class. Call input, process and output.
- **input** – declare a `Scanner` object and read two integers (the dividend and the divisor in that order) from the keyboard.
- **process** – calculate the quotient and remainder of performing integer division (dividend divided by divisor).
- **output** – display the results on the console screen.

SAMPLE KEYBOARD INPUT

Enter two integers: **25 4**

SAMPLE OUTPUT

25 divided by 4 equals 6 with a remainder of 1.

Lab 05B - 70 points

OBJECTIVE

WAP that reads three integers from the keyboard. Output the sum of the tens position and the sum of the ones position. For example: ***31***, ***44*** and ***12*** would result in **8** tens (*3 + 4 + 1 = 8*) and **7** ones (*1 + 4 + 2 = 7*).

FIELD SUMMARY

- **int a** – the first number read from the keyboard.
- **int b** – the second number read from the keyboard.
- **int c** – the third number read from the keyboard.
- **int tens** – value of the tens digit.
- **int ones** – value of the ones digit.

METHOD SUMMARY

- **main** – instantiate an object of your class. Call input, process and output.
- **input** – declare a `Scanner` object and read three integers from the keyboard using an appropriate prompt.
- **process** – Using integer and modulus division extract the value of the tens digits and the value of the ones digit for each of the three individual numbers. Store the sum of the tens digits in tens and the sum of the ones digits in ones.
- **output** – display the results on the console screen.

SAMPLE KEYBOARD INPUT

Enter a three integers: **31 44 12**

SAMPLE OUTPUT

Tens = 8
Ones = 7

SAMPLE KEYBOARD INPUT

Enter a three integers: **74 55 38**

SAMPLE OUTPUT

Tens = 15
Ones = 17

Lab 05C - 80 points

OBJECTIVE

WAP to input the number of seconds from the keyboard. Convert the number of seconds to the number of hours, minutes and remaining number of seconds that value represents. Use **integer division** to determine the number of minutes and **modulus division** to determine the number of seconds.

FIELD SUMMARY

- **int time** – the number of seconds read from the keyboard.
- **int hours** – the number of `hours` in **`seconds`**.
- **int minutes** – the number of `minutes` in **`seconds`**.
- **int seconds** – the number of `seconds` remaining after removing the number of hours and minutes.

METHOD SUMMARY

- **main** – instantiate an object of your class. Call input, process and output.
- **input** – declare a `Scanner` object and read an integer (the number of seconds) from the keyboard using an appropriate prompt.
- **process** – Using integer and modulus division break the number of seconds into hours, minutes and remaining seconds storing the resulting values in `hours`, `minutes`, and `seconds`.
- **output** – display the results on the console screen.

SAMPLE KEYBOARD INPUT

Enter the number of seconds: **31987**

SAMPLE OUTPUT

31987 seconds = 8 hours 5 minutes and 7 seconds.

Lab 05D - 90 points

OBJECTIVE

WAP to read a four digit number from the keyboard. Output the thousands, hundreds, tens, and the ones digits separately. Print each digit on a separate line with a label indicating its place value. Use **/** and **%** operators to separate the digits.

FIELD SUMMARY

- **int number** – the number read from the keyboard.
- **int thousands** – value of the thousands digit.
- **int hundreds** – value of the hundreds digit.
- **int tens** – value of the tens digit.
- **int ones** – value of the ones digit.

METHOD SUMMARY

- **main** – instantiate an object of your class. Call input, process and output.
- **input** – declare a `Scanner` object and read an integer from the keyboard using an appropriate prompt.
- **process** – Using integer and modulus division break the four digit number into its parts (*thousands*, *hundreds*, *tens*, and *ones*) storing each in a separate instance variable.
- **output** – display the results on the console screen.

SAMPLE KEYBOARD INPUT

Enter a four digit integer: **4528**

SAMPLE OUTPUT

4528 equals
4 thousands
5 hundreds
2 tens
8 ones

Lab 05E - 100 points

OBJECTIVE

WAP to convert an input value, such as 38.92 (3892 pennies), into the smallest number of bills and coins (20's, 10's, 5's, etc.). Do NOT use half-dollars. You must use **/** and **%**.

FIELD SUMMARY

- **double amount** – the amount of money read from the keyboard.
- **int cents** – the amount of money converted into an integer value.
- **int twenties** – the number of twenty dollar bills.
- **int tens** – the number of ten dollar bills.
- **int fives** – the number of five dollar bills.
- **int ones** – the number of one dollar bills.
- **int quarters** – the number of quarters.
- **int dimes** – the number of dimes.
- **int nickels** – the number of nickels.
- **int pennies** – the number of pennies.

METHOD SUMMARY

- **main** – instantiate an object of your class. Call input, process and output.
- **input** – declare a `Scanner` object and read a double from the keyboard using an appropriate prompt.
- **process** – break the four digit number into its parts (20's, 10's, 5's, etc.) storing each in a separate instance field.
  1. Convert the *amount* into an int value and store it in *cents* (2.52 = 252 pennies). You don't want to lose the fraction so the conversion process needs to multiply *amount* by 100 and type cast the result into an int.
  2. Use integer division to determine the number of $20 bills. Since the original amount was multiplied by 100 then the value of a 20 dollar bill must also be multiplied by 100 – hence, a 20 dollar bill is represented by 2000 (20 * 100).
  3. Use modulus division to remove the $20 dollar bills.
  4. Repeat steps 2 and 3 for the remaining values ($10, $5, $1, .25 cents, etc.)
- **output** – display the results on the console screen.

SAMPLE KEYBOARD INPUT
Enter a monetary amount: **38.92**
SAMPLE OUTPUT
38.92 = 3892 cents **1 = twenties** **1 = tens** **1 = fives** **3 = ones** **3 = quarters** **1 = dimes** **1 = nickels** **2 = pennies**