

CIS 162 Lab 2

Conversions

Preparation

Do the following before arriving at lab:

- Read chapter 3.1 – 3.7
- Read 1.6 with an emphasis on creating and using Scanners
- Read 3.7 with an emphasis on integer division and the remainder operator

Objectives

After completing this lab, you should be able to:

- *create* a project and class
- *read* data from the keyboard using the Scanner class
- *make calculations* using assignment statements and variables
- *display results* using the `print()` and `println()` methods

Overview

We talked tonight in lecture about variables and how to declare them. We saw that the data type modifier determines the type of data that can be stored in a variable. For instance, consider the following code:

```
import java.util.Scanner;

public class ReadInt {

    public static void main(String[] args){
        Scanner sc = new Scanner(System.in);
        int number;
        System.out.println("Please enter a number: ");
        number = sc.nextInt();
        System.out.println("You entered " + number + ".");
    }
}
```

This code will compile, but it will not run correctly if you attempt to enter a character or a floating point number. The Scanner will not be able to find an int and will crash (technically it will throw an exception but we won't learn about those until later!).

Consider this code though:

```
public class TestVars {
    public static void main(String[] args){
        int x = 42;
        int y = 1701;
        int z = y / x;
        float a = y / x;
```

```

        System.out.println("z = " + z);
        System.out.println("a = " + a);
    }
}

```

This may not print out what you expect! Why?

We may expect the value of a to be 40.5, but we would be wrong. The reason is that the values on the right side of an expression are evaluated before they are assigned to the left. So Java takes an int, divides it by an int which results in an int. It then stores the float representation of that int in the variable a giving us 40.0.

What about this:

```

public class TestVars2 {
    public static void main(String[] args){
        float x = 42;
        float y = 1701;
        float z = y / x;
        int a = (int)y / (int) x;

        System.out.println("z = " + z);
        System.out.println("a = " + a);
    }
}

```

This code sample probably is more what we expect. We get the value of 40.5 for z. For a we get 40. We needed to add the “(int)” code in front of our variables because Java won’t allow us to convert from a float to an int; it considers it an unsafe operation. In general Java won’t allow you to reduce the precision of variables unless you specifically tell it to. Adding precision Java will usually perform without question though. This is why we have no issues assigning an int to a float. Java can convert it with no potential loss of information.

Lab Activity #1 – Convert to seconds

1. Start BlueJ
2. Open the Project from last week (or create a new one).
3. Create a new Class called “ConvertToSeconds”
4. Edit the top comment section to describe the class and add both of your names as authors.
5. Remove the two automatically generated methods.
6. Create a main method (copy from Figure 1.6.2 in zyBook)
7. Prompt the user to enter three integers for hours, minutes and seconds.
8. Use variables with meaningful names
9. Calculate and display the number of equivalent seconds
10. When working correctly, copy to the corresponding zyLab in Chapter 3 for testing

Sample Output

Hours: 1

Minutes: 6
Seconds: 6
3966 seconds

Lab Activity #2 – Convert from seconds

1. Create a new Class called “ConvertFromSeconds”
2. Prompt the user to enter one integer for the number of seconds.
3. Calculate and display the number of equivalent hours, minutes and seconds (refer to section 3.7 about integer division and the modulo or remainder operator)
 `hours = total / 3600;`
 `total = total % 3600;`
4. When working correctly, copy to the corresponding zyLab in Chapter 3 for testing

Sample Output

Seconds: **20000**

This is 5 hours, 33 minutes and 20 seconds.

Lab Activity #3 – Convert to dollars

1. Create a new Class called “ConvertToDollars”
2. Prompt the user to enter four integers for quarters, dimes, nickels and pennies.
3. Calculate the total number of cents by combining quarters, dimes, nickels and pennies.
4. Divide by 100.0 and place in **a double**
5. Display the equivalent in dollars and cents.
6. When working correctly, copy to the corresponding zyLab in Chapter 3 for testing

Sample Output

Quarters: **6**

Dimes: **6**

Nickels: **6**

Pennies: **6**

This is equivalent to \$2.46

Lab Activity #4 – Mad Lib

A mad lib is a simple game where you ask someone to name a few specific types of words and then you create a paragraph by inserting the words. The result is usually pretty silly but really fun for ten year olds!

1. Find a mad lib on the Web or create one on your own.
2. Create a new class called MadLib with a main method.
3. Prompt the user for several words. Your mad lib should have at least five requested words. Refer to section 4.5 about reading Strings.
4. When working correctly, show your instructor

Simple Sample

Enter a food: **hamburger**

Enter a color: **red**

Enter an emotion: **sad**

Eating red hamburger makes me sad.

Lab Activity #5 – Four 4s Problem

Use the number 4 four times in various permutations to create expressions that result in each value from 0 – 9. Use integer division and any of the four operators and parens (e.g. +, -, *, /). There may be more than one solution for each result. Show your instructor when complete.

$$0 = 4 + 4 - 4 - 4$$

$$1 = (4 + 4) / (4 + 4)$$

$$2 = (4 / 4) + (4 / 4)$$

$$3 =$$

$$4 =$$

$$5 =$$

$$6 =$$

$$7 =$$

$$8 =$$

$$9 =$$