

Reading:

Chapter 2.1 - 2.6 pages 58-91

Terminology:

string literal	assignment statement	arithmetic operator
method	assignment operator	operator precedence
parameter	constant	increment / decrement operators
string concatenation	byte, short, int, & long	assignment conversion
escape sequence	float & double	promotion
variable	char	casting
variable declaration	boolean	delimiter
variable initialization	character literal	

COMP 1210 Coding Standard Additions

- Follow the standards in the book when naming variables and constants.
 - Variables and constants should be declared at the beginning of the main method.
 - Variables should start with a lower case and should use medial capitals (also called “camelCase”). Examples:
numberPeople, xAxis, result
 - Constants should be in all caps with words separated by an underscore. Examples:
MAX_NUMBER, BASELINE, TAX_RATE
 - Both variables and constants should have descriptive variable names. **This convention is not enforced by Checkstyle; however, it will be graded by your GTA in the projects.**
Example:
Good 😊: `int numberJobs, numberEmployees;`
Bad ☹: `int num1, num2;`

Due:

Activity (in-lab): Monday, August 27, 2012 by the end of lab

Goals:

By the end of this activity you should be able to do the following:

- Get user input using the Scanner class
- Create variables and concatenate variables with a string
- Use arithmetic operators to manipulate **integer** and **double** data types
- Use a basic *if* statement to determine which output is appropriate

Program Description:

The goal of this activity is to create a program that displays someone's name, their age, and calculations based on their age.

In-lab Directions:**Part 1: Getting user's name and age (25%)**

- Create the following class in jGRASP and
Save as **H:/COMP1210/Lab2/AgeStatistics.java**

```
import java.util.Scanner;

public class AgeStatistics {
    public static void main(String args[]) {
    }
```

- Declare the following variables in the main method:

```
String name;
int ageInYears;
int gender = 0; // 1 for female
double maxHeartRate = 0;

Scanner userInput = new Scanner(System.in);
```

- Prompt the user for their name:

```
System.out.print("Enter your name: ");
name = userInput.nextLine();
```

- Prompt the user for their age:

```
System.out.print("Enter your age in years: ");
ageInYears = userInput.nextInt();
```

- Add lines to prompt for and input the user's gender under the previous code. Make sure that your program has the EXACT output as below.

If text is bold, then you have to come up with the code on your own!

```
----jGRASP exec: java AgeStatistics

Enter your name: Lauren Goff
Enter your age in years: 70
Enter your gender (1 for female, 0 for male): 1

----jGRASP: operation complete.
```

Part 2: Converting age (25%)

- First, calculate and display the age of the user in minutes (minutes = years * 525600)

```
// convert age
System.out.println("\t Your age in minutes is " +
    ageInYears * 525600 + " minutes.");
```

- Now calculate the age in centuries. Make sure that you can tell the differences between the various types of conversion (pg 84 in the book) for the quiz on Wednesday.

```
System.out.println("\t Your age in centuries is " +
    (double) _____ / 100 + " centuries.");
```

```
----jGRASP exec: java AgeStatistics

Enter your name: Lauren Goff
Enter your age in years: 71
Enter your gender (1 for female, 0 for male): 1
    Your age in minutes is 37317600 minutes.
    Your age in centuries is 0.71 centuries.

----jGRASP: operation complete.
```

Part 3: Max heart rate prompt (35%)

- Add a line to print max heart rate:

```
// display max heart rate
System.out.print("Your max heart rate is ");
```
- Add an *if* statement to print a different max heart rate calculation for male and female. The comments below are there to help you in the Part 4 where you will insert the code to do the calculations on the line after each comment.

```
if (gender == 1) { // calculate female MHR on next line
}
else { // calculate male MHR on next line
}
}
```

- Add one more print statement to print the max heart rate:
`System.out.println(maxHeartRate + " beats per minute.");`

Part 4: Max heart rate calculations (15%)

- On your own, complete the *if* statement to calculate the max heart rate calculation for females and for males. The max heart rate is calculated as follows:
 - Max heart rate for females: $209 - (0.7 * \text{age})$
 - Max heart rate for males: $214 - (0.8 * \text{age})$

Test your program and make sure that the output is exactly as shown below. Your lab instructor may ask you to use a different set of values to test your program when you demo it for your activity grade.

Output # 1:

```
----jGRASP exec: java AgeStatistics

Enter your name: Lucie
Enter your age in years: 25
Enter your gender (1 for female, 0 for male): 1
    Your age in minutes is 13140000 minutes.
    Your age in centuries is 0.25 centuries.
Your max heart rate is 191.5 beats per minute.

----jGRASP: operation complete.
```

Make sure your
output for this
number is correct!

Output # 2:

```
----jGRASP exec: java AgeStatistics

Enter your name: Kao
Enter your age in years: 19
Enter your gender (1 for female, 0 for male): 0
    Your age in minutes is 9986400 minutes.
    Your age in centuries is 0.19 centuries.
Your max heart rate is 198.8 beats per minute.

----jGRASP: operation complete.
```

Make sure your
output for this
number is correct!