



Project 2: Using Variables and Expressions



Project 2 Overview

- For this project you will work with three programs
 - Circle
 - Paint
 - Ideal_Weight
- What you need to do for each program is described on the following slides.



Circle.java

- An initial version of Circle.java is available for download.
- Create a folder named Circle in a convenient location.
- Download the file Circle.java from the Downloads area of the class web site to your Circle folder.

http://www.cse.usf.edu/~turnerr/Programming_Concepts/Downloads/Project_2

- Study Circle.java and try to understand how it works.
 - See next slide

Some things to notice:

- The first three lines inside *main* are declarations for *PI*, *radius*, and *area*. Note that the type for each is given in these lines: *final double* for *PI*, since it is a floating point constant; *int* for *radius*, since it is an integer variable, and *double* for *area*, since it will hold the product of the radius and *PI*, resulting in a floating point value.
- These first three lines also hold initializations for *PI*, *radius*, and *area*. These could have been done separately, but it is often convenient to assign an initial value when a variable is declared.
- The next line is simply a print statement that shows the area for a circle of a given radius.
- The next line is an assignment statement, giving variable *radius* the value 20. Note that this is not a declaration, so the *int* that was in the previous *radius* line does not appear here. The same memory location that used to hold the value 10 now holds the value 20—we are not setting up a new memory location.
- Similar for the next line—no *double* because *area* was already declared.
- The final print statement prints the newly computed area of the circle with the new radius.



Add Circumference to Circles Program

- The circumference of a circle is two times the product of pi and the radius. Add statements to this program to compute the circumference in addition to the area for both circles.
- You will need to do the following:
 - Declare a new variable to store the circumference.
 - Compute the circumference for each value of the radius and store it in that variable.
 - Add two println statements to show your results. Be sure your results are clearly labeled.
 - Compile and test the modified program.



Compute Change

- When the radius of a circle doubles, what happens to area and circumference? Do they double as well?
- You can determine this by dividing the second area by the first area. Unfortunately, as it is now the program overwrites the first area and circumference with the second area and circumference. You need to save the first area and circumference you compute instead of overwriting them with the second set of computations. So you'll need two area variables and two circumference variables, which means they'll have to have different names (e.g., area1 and area2). Remember that each variable will have to be declared.



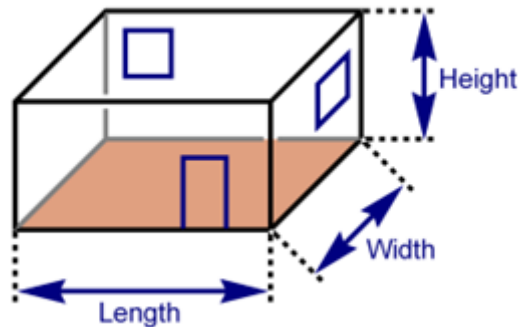
Compute Change

Modify the program as follows:

- Add declarations for the area and circumference of the second circle.
- Use the new variables when calculating area and circumference of the second circle.
- Change the println statements for the second circle to use the new variables.
- At the end of the program, compute the area change by dividing the second area by the first area. This gives you the factor by which the area grew. Store this value in an appropriately named variable (which you will have to declare).
- Add a println statement to print the change in area that you just computed.
- Now repeat the last two steps for the circumference.
- Save the file, compile, and run it.
- Correct mistakes if necessary.

Painting a Room

File Paint.java, available in the Downloads area of the class web site, contains a partial program, which when complete will calculate the amount of paint needed to paint the walls of a room of the given length and width. It assumes that the paint covers 350 square feet per gallon.





Paint.java

- Create a folder named Paint in a convenient location.
- Download the file Paint.java from the Downloads area of the class web site to your Paint folder.
http://www.cse.usf.edu/~turnerr/Programming_Concepts/Downloads/Project_2
- Fill in the missing statements so that the program does what it is supposed to. (Comments tell you what to fill in where.)
- Compile and run the program and correct any errors.
- Suppose the room has doors and windows that don't need painting. Ask the user to enter the number of doors and number of windows in the room, and adjust the total square feet to be painted accordingly. Assume that each door is 20 square feet and each window is 15 square feet.
- Again compile and test the program.



Ideal Weight

- Write a program to compute the ideal weight for both males and females who are over 5 feet tall.
- According to one study, the ideal weight for a female who is over 5 feet tall is 100 pounds plus 5 pounds for each inch in height over 5 feet. For example, the ideal weight for a female who is 5'3" would be $100 + 15 = 115$ pounds.
- For a male who is over 5 feet tall, the ideal weight is 106 pounds plus 6 pounds for each inch in height over 5 feet. For example, the ideal weight for a male who is 6'2" would be $106 + 14 \times 6 = 190$ pounds.
- Your program should ask the user to enter his/her height in feet and inches (both as integers—so a person 5'3" would enter the 5 and then the 3). It should then compute and print both the ideal weight for a female and the ideal weight for a male.



Ideal Weight

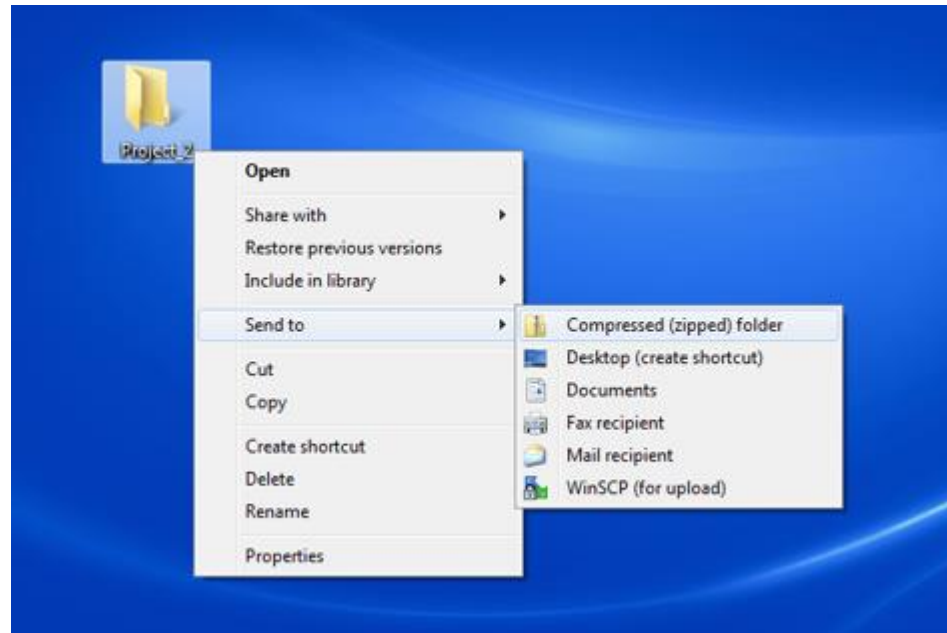
The general outline of your main function would be as follows:

- Declare your variables.
 - Think about what variables you need—you need to input two pieces of information, then you need some variables for your calculations. (See the following steps.)
- Get the input (height in feet and inches) from the user.
- Compute the total number of inches of height. (Convert feet and inches to total inches).
- Compute the ideal weight for a female and the ideal weight for a male. (Here you basically convert the "word" description above to assignment statements).
- Print the answers.

Plan your program, then type it in, compile and run it. Be sure it gives correct answers.

Submission

- Put your three Java source files into a folder and zip the folder.
 - In Windows, right click on the folder, select "Send to" then select "Compressed (zipped) folder".





Submission

- Submit your zipped Java source files via Canvas Assignments.
- Project is due by 11:59 PM
 - Monday, January 25 All Sections



Ground Rules

- Do not share your code with other students
 - Before or after submitting the project.
 - OK to *discuss* the project.
- Do not copy any other student's code.
 - Or even look at it.
- Do not let anyone copy or examine your code.