**Activity: Fun Factorials**

In this activity, you will write a **for**() loop, a **while**() loop and a **do**-**while**() loop to calculate the factorial of a given number. You will also use an **if**() statement to check to see if the user enters valid data.

**Project Details**

A *factorial* is the product of all integers less than or equal to a value and greater than zero. For example, the factorial of the number 5 is result of multiplying together 5, 4, 3, 2, and 1: **5!** = 5 x 4 x 3 x 2 x 1 = 120.

Your program will ask the user for a number and will then use each type of loop to calculate the factorial.

**How to Complete this Activity**

Now that you have gained some programming experience, our activity descriptions will contain a little less detail (such as specifying every single variable name), and you will write your own code according to the requirement. If you are ever confused about certain tasks, don’t forget to check the course activity solution.

Here are the steps you need to take to complete this activity.

1. • Create a new project in Eclipse called “Factorials” and add a new class called **Factorials**.
2. • In your **main**() function, create a **Scanner** variable to gather user input
3. • Create an integer variable to hold the user input
4. • Print a message to the screen asking the user to enter a number

• Use a **while**() loop to verify the user has entered valid data. *While* the **Scanner hasNextInt**() method returns **false**:

* 1. o Print a message telling the user to try again.
  2. o Use **nextLine**() within your loop to flush the input stream, including the newline character, so the user can try again

• Use the **Scanner nextInt**() method to pull the integer from the input stream and store in your integer variable.

• Declare another integer variable to hold the result of the factorial calculation and initialize it to 1.

• Use a for() loop to calculate the factorial of the user’s number and print the result to the user.

o Your loop index should range from 1 up to and including the user’s input value

* + 1. o Multiply the current result variable by the loop index
    2. o Print out the result using System.out.println().
  1. • Use a while() loop to calculate the factorial of the user’s number and print the result to the user.
     1. o Use an approach similar to the for() loop.
     2. o Initialize the result to 1, and then create another index variable that you will increment by 1 each time through the loop. Use the index to multiply into the result value.
     3. o Print out the result using System.out.println().
  2. • Use a do-while() loop to calculate the factorial of the user’s number and print the result to the user.
     1. o Use an approach similar to the while() loop.
     2. o Print out the result using System.out.println().

Activity Output

Here is some example output for your program. The exact strings you print to the screen are up to you. But you should ensure that each loop produces the same, correct factorial result.

Enter an integer number and press Enter: a

Oops! Try again: 5

for() loop result: 120

while() loop result: 120

do-while() loop result: 120

Notice that if you enter an integer larger than 16 or so, the resulting factorial will be so large that an integer result variable cannot hold it! This example shows the resulting calculation with an integer result. See how we “wrapped around” into negative numbers?

Enter an integer number and press Enter: 17

for() loop result: -288522240

while() loop result: -288522240

do-while() loop result: -288522240

So you may want to use a long data type to hold a larger result.

Even then, factorial results increase very quickly, so you will find a maximum input beyond which a long won’t work either! 17! will work:

Enter an integer number and press Enter: 17

for() loop result: 355687428096000

while() loop result: 355687428096000

do-while() loop result: 355687428096000

But still, 21! will not fit into a long result either.

Enter an integer number and press Enter: 21

for() loop result: -4249290049419214848

while() loop result: -4249290049419214848

do-while() loop result: -4249290049419214848

If you want to be extra-safe in your program, you can check the user’s input for a value that is too high to produce a correct result.