# Hands-on Experiment # 3: Worksheet

Section\_\_\_\_\_\_1\_\_\_\_\_\_\_\_ Date\_\_\_\_\_\_\_\_\_\_\_25/8/2563\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID \_\_\_\_\_\_6338110221\_\_\_\_\_\_\_\_\_\_\_\_ Name\_\_\_\_\_\_\_\_Nonthapat Kewamporn\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## Part A: Data Type & Type Conversion

Fill out the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Express | Data Type | Result | Reason |
| 3/4 | int | 0 | Both operands are integer. |
| 40L+7F | float | 47.0 | Long is converted to float |
| 9.2%6 | double | 3.1999999999999993 | Int is converted to double |
| 4.5%3.1 | double | 1.4 | Both operands are double |
| 13%5 | int | 3 | Both operands are integer |
| -3%2 | int | -1 | Both operands are integer |
| 9+(double)4 | double | 13.0 | Int is converted to double |
| 1.5f+3 | float | 4.5 | Int is converted to float |
| (int)5.7+4.5 | double | 9.5 | Int is converted to double |
| (double)5+“6” | String | 5.06 | Double is converted to String |
| (int)3.4+4.2f-3d | double | 4.199999809265137 | Int is converted to float, float is converted to double |

Study the method “getDataType” in the program “DataTypeExample.java”. Modify the program to check the types of expressions in the table.

public class DataTypeExample {

public static String getDataType(Object obj){

return ((Object) obj).getClass().getName().toString();

}

public static void main(String[] args) throws Exception{

System.out.println(getDataType(3/4));

}

}

## Part B: Increment/Decrement Operators

Look at the following code.

public class Challenge2 {

public static void main(String[] args) {

int x = 0;

x = x++;

System.out.println("A: " + x);

System.out.println("B: " + x++);

}

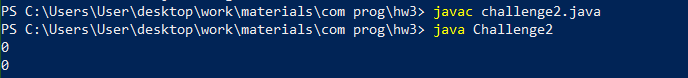
}

What would be the program output? (Determine without running the program)

A: 0

B: 0

Run the program. Capture output.



Explain why the program show that results.

- X is declared as int and assigned to be 0

- X value is given to itself, so x = 0 again. However, the previous x is incremented but it does not affect the new value of x whatsoever. So, x’s value is ultimately 0

- print the first line, which is the value of x. So, the line prints out 0

- print the second line, which is the value of x before incrementing. So, the line prints out 0, but the value of x now 1

## Part C: Formatting Output

Java library provides the value of the number *e*, known as *Euler's number*, for you to use as Math.E. Use System.out.format() to format the output as shown below. (You can look at “<https://alvinalexander.com/programming/printf-format-cheat-sheet>” for the format string and how to use it.

A picture containing drawing, table

Description automatically generated

|  |  |
| --- | --- |
| Show the value of Math.E as | Format used in System.out.format() |
| 2.718281828459045 | %s |
| 2.718282 | %.6f |
| 2.7183e+00 | %e |
| 2.72e+00  *(note: display in right align to the width of the screen of 80 characters)* | %80.2e |
| +2.718  *(note: occupies 8 characters)* | %+8.3f |
| 2.7138 | %.4f |
| 002.7138 | %08.4f |
| 2.7138 | %8.4f |

Write a java program that uses System.out.format() to show the *e* value according to the previous table.

Copy your code and put in the space below.

System.out.format("%s\n",Math.E);

System.out.format("%.6f\n",Math.E);

System.out.format("%e\n",Math.E);

System.out.format("%80.2e\n",Math.E);

System.out.format("%+8.3f\n",Math.E);

System.out.format("%.4f\n",Math.E);

System.out.format("%08.4f\n",Math.E);

System.out.format("%8.4f\n",Math.E);

## Part D: User Input/output and complex expression

Factorial of n can be estimated by the following formula:

|  |  |
| --- | --- |
| Stirling (simple) |  |
| Stirling (lower-bound) |  |
| Stirling (upper-bound) |  |

1. Use provided excel worksheet (FactorialEstimation.xlsx) to calculate and fill out the following results (change the format to show up to 15 digits after decimal point). If the value is to large, show it as scientific number.

|  |  |  |  |
| --- | --- | --- | --- |
| Dataset # | Stirling(simple) | Stirling(lower-bound) | Stirling(upper-bound) |
| 1 | 0.922137008895789 | 0.995870161462797 | 1.002274449182227 |
| 2 | 1.919004351488983 | 1.997320404751810 | 2.000652047690966 |
| 5 | 118.019167957590100 | 119.969853959208900 | 120.002637086196980 |
| 10 | 3598695.618741037300000 | 3628560.141985111000000 | 3628810.051426933000000 |
| 20 | 2.422786846761135e+18 | 2.432860790060010e+18 | 2.432902852332159e+18 |
| 50 | 3.036344593938168e+64 | 3.041400953459955e+64 | 3.041409387750493e+64 |
| 100 | 9.324847625269420e+157 | 9.332615094728998e+157 | 9.332621570317666e+157 |
|  |  |  |  |

Complete the given program called FactorialApproximation.java that take ONE argument for input as n, then calculate the factorial approximation using the Stirling formula’s given in the table above.

The output of the program shall look like:

> **java FactorialApproximate 1**

The approximated value of 1! using Stirling:

simple: 0.9221370088957891

lower: %.15f0.9958701614627972

upper: %.15f1.0022744491822266

> **java FactorialApproximate 100**

The approximated value of 100! using Stirling:

simple: 9.324847625269393E157

lower: 9.332615094728998E157

upper: 9.332621570317666E157

Copy your code and put in the space below.

public class FactorialApproximate

{

public static void main(String[] args)

{

double n = Double.parseDouble(args[0]);

double simple = Math.sqrt(2\*Math.PI\*n) \* Math.pow(n/Math.E,n);

double lbound = Math.sqrt(2\*Math.PI) \* Math.pow(n,(n+0.5)) \* Math.pow(Math.E,(-n + 1/(12\*n+1)));

double ubound = Math.sqrt(2\*Math.PI) \* Math.pow(n,(n+0.5)) \* Math.pow(Math.E,(-n + 1/(12\*n)));

System.out.format("The approximated value of %d! using Stirling:\n",(int) n);

System.out.format("Simple: %.15f\n",simple);

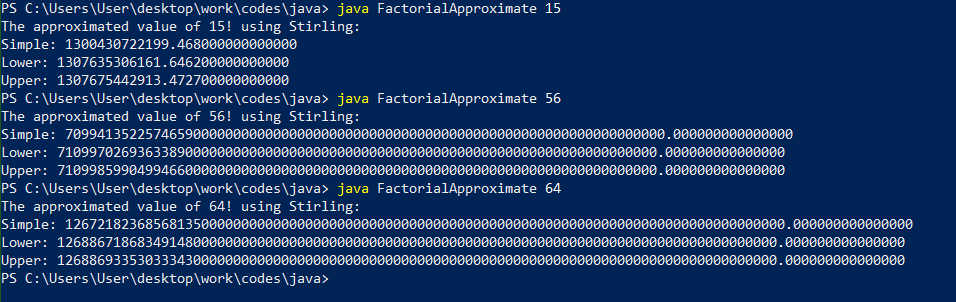
System.out.format("Lower: %.15f\n",lbound);

System.out.format("Upper: %.15f\n",ubound);

}

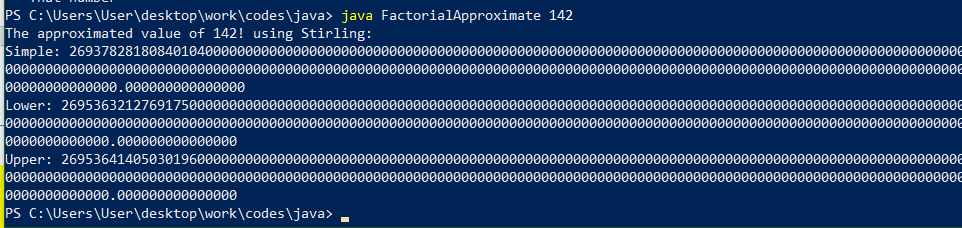
}

1. Capture your output of the program using your own data (at least 3 values between 10 to 100).

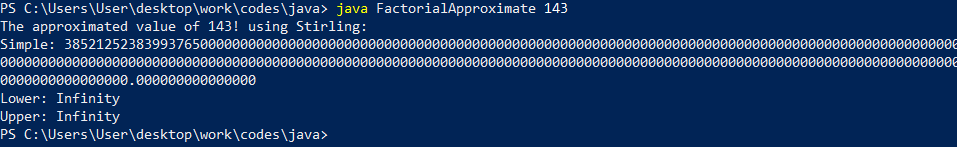


1. What is the maximum integer value that the program can calculate all three approximations before it shows “Infinity”. Capture your output of your program.

142



If 143, this happens



## Part E: Code Review/Reading & Problem Solving

The following program takes four inputs, your name, your age, your gpa and your major, then display those information.

import java.util.Scanner;

public class KeyboardInput {

public static void main(String[] args) {

Scanner kbd = new Scanner(System.in);

System.out.print("What is your name? ");

String a = kbd.nextLine();

System.out.print("How old are you? ");

int b = kbd.nextInt();

System.out.print("What is your high school's GPA? ");

double c = kbd.nextDouble();

System.out.print("What is your major in ISE? ");

String d = "N/A";

d = kbd.nextLine();

System.out.format("My name is [%s].%n", a);

System.out.format("I am [%d] years old.%n", b);

System.out.format("I graduated with [%f] GPA.%n", c);

System.out.format("Currently, I study [%s] in ISE.%n", d);

}

}

1. Review the code for readability. Write down what need to be changed in the space below.

- Remove all the [ ]

- change the variable name according to their purpose: name, age, gpa, major

- For the System.out.format("I graduated with [%f] GPA.%n", gpa)

change %f to %.2f

- For the System.out.format("Currently, I study [%s] in ISE.%n", major),

change "Currently, I study [%s] in ISE. To “Currently, I’m majoring in %s in ISE”

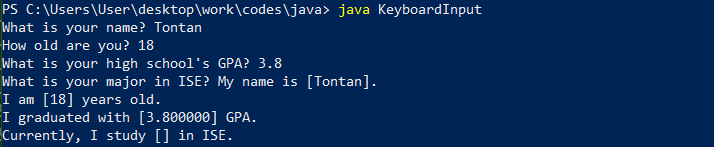
1. What are the functionality of kbd.nextLine(), kbd.nextInt(), and kbd.nextDouble().

Kdb.nextLine() scans for user input when beginning next line(in default, saves value as String)

Kdb.nextInt() scans for user input and saves it as integer

Kdb.nextDouble() scans for user input and saves it as double

1. Change your program according to what need to be changed in (1) and run the program with your real data. Capture the output and paste in the space below.



1. Is the result as you expected? Why?

No, since nextInt and nextDouble method doesn’t read the new line after we press enter. We never explicitly told the program to do so, which is why the question “What is your major in ISE” doesn’t prompt for input.

1. Fix your program to get the result as you expected. Copy your code and paste in space below.

import java.util.Scanner;

public class KeyboardInput {

public static void main(String[] args) {

Scanner kbd = new Scanner(System.in);

System.out.print("What is your name? ");

String a = kbd.nextLine();

System.out.print("How old are you? ");

int b = kbd.nextInt();

kbd.nextLine();

System.out.print("What is your high school's GPA? ");

double c = kbd.nextDouble();

kbd.nextLine();

System.out.print("What is your major in ISE? ");

String d = "N/A";

d = kbd.nextLine();

System.out.format("My name is [%s].%n", a);

System.out.format("I am [%d] years old.%n", b);

System.out.format("I graduated with [%f] GPA.%n", c);

System.out.format("Currently, I study [%s] in ISE.%n", d);

}

}

1. Explain how do you fix the problem.

I added kbc.nextLine(); after any line that has kbc.nextInt(); or kbc.nextDouble(); So that the program could prompt for read the ne

Submit this worksheet (by only one member of the group) via <http://www.myCourseVille.com> (Assignments > Assignment 3) before midnight of three days after your lecture.