Ira A. Fulton Schools of Engineering
Arizona State University

school of computing, informatics, & decision systems engineering

CSE 110 – Assignment #2

Maximum points: 20 pts

Topics

- If statements (`if, if/else, if/else if`)
- Expressions (<, >, ==, !=)
- **String** methods: `indexOf (String), substring(int, int), compareTo(String), toUpperCase(), toLowerCase()`

Use the following Guidelines:

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- User upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.

Important Note:

All submitted assignments must begin with the descriptive comment block. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment. Your programming assignments require **individual** work and effort to be of any benefit. Every student must work independently on his or her assignments. This means that every student must ensure that neither a soft copy nor a hard copy of their work gets into the hands of another student. Sharing your assignments with others in any way is **NOT** permitted. Violations of the University Academic Integrity policy will not be ignored. The university academic integrity policy is found at
http://www.asu.edu/studentlife/judicial/integrity.html

Part 1: Writing Exercise: (5 pts)

This assignment has two parts. The first part is to write the answers briefly in your words. and it is not a coding question but English writing question. Part 1 is a hint for Part 2. Part 1 shows a sample code to try to sort three integers, and Part 2 is asked to make a program to sort three strings. Try to answer the Part 1 questions first, then apply to the string case in Part 2.

The following code is supposed to read three integers and **sort them from the smallest to the largest.** It creates six integer variables. The **`a, b`**, and **`c`** are used for the first, second, and third user inputs. The **`small`**, **`middle`**, and **`large`** are used to assign the values from **`a, b, c`** corresponding to the order of values. Write the code and test it. If it works, then answer each question below. (Don't submit this ThreeInteger.java as an assignment #2)

```
1  import java.util.Scanner;
2  public class ThreeInteger {
3      public static void main (String[] arg){
4            Scanner scanner = new Scanner(System.in);
5            int a=0; int b=0; int c=0; int small = 0; int middle = 0; int large = 0;
6            boolean isValid=true;
7
8            System.out.println("Input the first integer");
9            if(scanner.hasNextInt() && isValid) a = scanner.nextInt();
10           else isValid = false;
11           System.out.println("Input the second integer");
12           if(scanner.hasNextInt() && isValid) b = scanner.nextInt();
13           else isValid = false;
14           System.out.println("Input the third integer");
15           if(scanner.hasNextInt() && isValid) c = scanner.nextInt();
16           else isValid = false;
17
18           if(isValid){
19              if (b - a > 0) { small = a; large = b;}
20              else{ small = b; large = a; }
21              if ( c - small < 0){ middle = small; small = c;}
22              else if (c - large > 0){ middle = large; large = c;}
23              else{ /*Need a statement;*/}
24              System.out.printf("The numbers are %d %d %d.\n", small, middle, large);
25           }
26           else  System.out.println("The input was invalid");
27     }
28 }
```

a. This program is **not completed** but keep it as it is. Test six cases to figure out what is wrong. For the case of 1, 4, and 6 inputs, we need to test the 6 cases of one by one. (first, second, third) = (1, 4, 6), (1, 6, 4), (4, 1, 6), (4, 6, 1), (6, 1, 4), and (6, 4, 1). Test all of them above and write the input and output of **wrong case(s),** and explain the reason in your word. **(1pt)**

b. In the line 23 (the line numbers are shown in the left of code), write a single statement in the else-block to make the program sort in all six cases correctly. **(1pt).**

c. The code is to compare three numbers.
Suppose to develop another program to compare three string data. When you check the

alphabet order of string data instead of numbers, it is required to use `compareTo(String)` method. For example, if you want to compare `str1="Yoshi" and str2="Mario"`, use

```
if(str1.compareTo(str2) > 0)
```

Read the textbook (Special Topic 3.2) and answer whether the if-statement above is evaluated as true or false, and explain the reason in your words. **(1 pt)**.

d. Answer the result of the snipped code below, and explain the reason. **(1 pt)**.

```
String a = "Yoshi";
String b = "yoshi";
System.out.println( a.compareTo(b) );
```

e. All characters are ordered in Java. The alphabet letters are ordered as "A", "B", "C"... "Z", "a", "b".... "y", "z". In other words, if a capitalized letter is not between "A" and "Z", the letter must not be an alphabet such as a special symbol or number. Make the if-statement to check if a string variable (supposed to be a single letter), **`str`**, is an alphabet letter or not. **(1 pt)**.

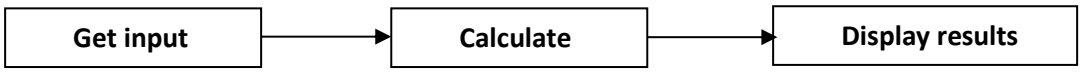
```
if (
    str.toUpperCase().compareTo(???) >=0 &&
    str.toUpperCase().compareTo(???) <= 0  )
```

Note: The answers to the 5 questions (a through e) above should be typed in the block of comments in the java file such as;

```
//*********************************************************
// Name: Your name here
// Title: AssignmentX.java
// Description: Short description should be typed here.
// Date: mm/dd/yyyy
//*********************************************************
/*
a) For Input (X, X, X), it returns a wrong output as (X, X, X),
because …
b) xxx = xxxx;
c) XXX because …
d) XXX because …
e) XXX and XXX
*/
```

Part 2: Programming (15 pts)

Write a Java program called **Assignment2.java**. The program is to display questions and read user inputs, then calculate and print out the requested value with a proper format. This program will follow a very simple process.
*) **Part1 is a big hint for this part.**



1. Read three names one by one, and display the list in alphabetic/lexicographic order **(5 pts)**. For example, when the inputs are "Smith", "john", and "mike", it displays:

   JOHN, MIKE, SMITH

2. All letters of output (displayed) name should be capitalized **(5 pts)**. Line breaks may be different from the submission output and your local output, but no problem.
3. However, if the name **starts with** non-alphabetic letter, then display an error message **(5 pts)** after it is input. In the case, use a blank name. Look at the example execution for more details and the display format.

   (*) Use the `compareTo(String)` method. (Special Topic 3.2 in Textbook)

IMPORTANT on Programming

- It is allowed to make only **one Scanner variable**. In your PC, it may work with multiple Scanner variables, but the server site (online submission) does not accept it. If you make more than one Scanner variables, **you may lose all 15 points**.
- Use only the Java statements that have been covered in class (or textbook) to date. This means you CAN use declaration, assignment, input and output statements. **DO NOT** use any other statements (loop, array, break, sort, etc.). If in doubt, ask your TA or instructor. If you use them, then you lose the points of task.
- Use the scanner's **`next()`** method instead of **`nextLine()`** when you read the String data using Scanner in this assignment. Your PC can work in either way, but not on the server site.

Example Execution:
The following is an example input and output. **The input is shown in red (Not displayed But typed)**. Make your own questions rather than this example.

Example 1

```
----------------
YOUR OUTPUT 1
----------------
```

```
*** TASK: Read name and display them in alphabetic order ***
Please input he first name: Smith
    SMITH
Please input the second name: jOHN
    JOHN, SMITH
Please input the third name: mike
    JOHN, MIKE, SMITH

*** END OF Assignment#2 ***
```

Example 2

```
----------------
YOUR OUTPUT 2
----------------
*** TASK: Read name and display them in alphabetic order ***
Please input he first name: 007James
    Error: The first letter should be an alphabet
Please input the second name: chris
    CHRIS
Please input the third name: GEORGE
    CHRIS, GEORGE

*** END OF Assignment#2 ***
```

/**********

Submit your homework by following the instructions below:

**********/

- Go to the course web site (my.asu.edu), and then click on the GradeScope on CANVAS.
- Submit your **Assignment2.java** file on-line in Assignment#2 item in Gradescope.
- The **Assignment2.java** should have the following, in order:
- In comments, the assignment Header including your name, title, lab, date, etc.
- In comments, the answers to questions presented in Part#1.
- The working Java code requested in Part #2.
- The file must compile and run as you submit it. You can confirm this by viewing your submission results.

Important Note: You may resubmit as many times as you like until the deadline, but we will only mark your last submission. **NO LATE ASSIGNMENTS WILL BE ACCEPTED.**