South Louisiana Community College
ASDV 1220, Programming Fundamentals

Work with same partner unless your instructor reassigns you to work with another partner! Use ONE computer together with your partner.

Learning Objectives

After completion of this lab, you should be able to

1. Understand the String calls to do input.
2. Understand the String method to extract characters
3. Understand the String method for lower and upper case conversions
4. Understand the Character class and the char type
5. Understand rounding

Create project Lab10

Problem 1

Create a class **ReadWriteName1**, write the code as shown below that simply reads the first and last name using the String method next(), and then prints them.

```
4  import java.util.Scanner;
5
6  public class ReadWriteName1
7  {
8      public static void main(String[] args)
9      {
10         Scanner scan = new Scanner( System.in );
11         System.out.print("Please enter your first name followed by your last name \nbeing separated "
12                 + " from each other by whitespace or carriage return: ");
13         String firstName = scan.next();
14         String lastName = scan.next();
15
16         System.out.println("\nHello " + firstName + " " +  lastName + "!");
17
18     }
19 }
```

Problem 2

Create a class **ReadWriteName2,** which prompts the user to enter his first name, then reads the first name, then prompts the user to enter his last name, then reads the last name. Prints the first and last name separated by comma.

Problem 3

Create a class **ReadWriteName3**, write the code as shown below that simply reads the first and last name using the String method nextLine() and then prints the line.

1.

```
6   public class ReadWriteName3
7   {
8        public static void main(String[] args)
9       {
10          Scanner scan = new Scanner( System.in );
11          System.out.print("Please enter your first name followed by your last name \nbeing separated "
12                  + " from each other by whitespace: ");
13          String wholeLine = scan.nextLine();
14
15          System.out.println("\nHello " + wholeLine + "!");
16
17      }
18  }
```

Problem 4

Create a class **ReadWriteCapsOrSmallLetters** and write code which prompts the user to enter 5 words in one line where each word separated by whitespace(s). Read the words and then reprint them by using all capitals, and then by using all small case letters. Use the String methods toUpperCase() and toLowerCase().

Example code:

String s = "ABC";

s = s.toUpperCase() ;

Converts string s to uppercase sand assigns to variable s. has value "abc".

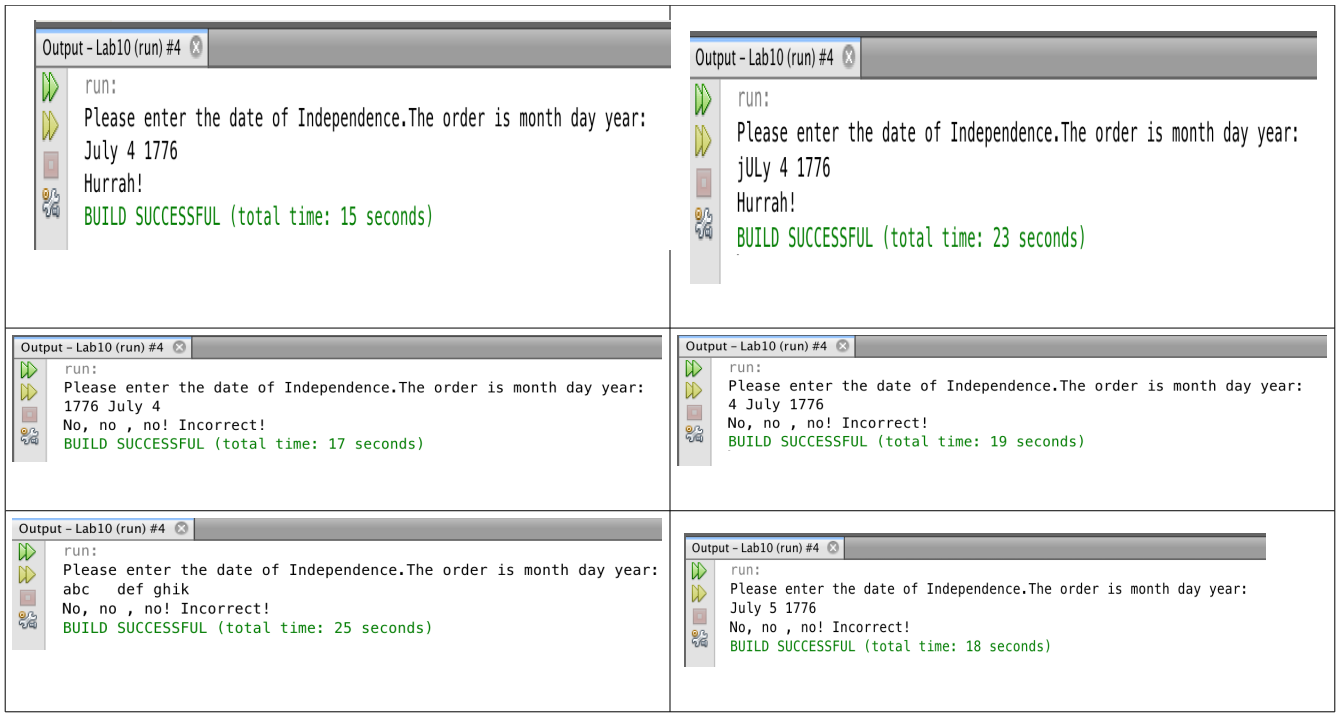
```
run:
Please enter 5 words in one line separated by whitespace and i will reprint them using capital and small case letters
Sunday Monday Tuesday Wednesday        Thursday Friday

The line you entered in all capital letters: SUNDAY MONDAY TUESDAY WEDNESDAY        THURSDAY FRIDAY

The line you entered in all small case letters: sunday monday tuesday wednesday        thursday friday
BUILD SUCCESSFUL (total time: 1 minute 1 second)
```

Problem 5

Create a class **IndependenceUSA** and write code which asks the user to enter text that contains the American Independence day (July 4th 1776). Read the date using the method **next**() 3 times. The user should enter the Independence day in the order month, day, year as shown in the first two runs below(we are not case sensitive) . Use the String method **compareToIgnoreCase()** to test the validity of the date and print "Hurrah!" if the date entered is correct, or "No, no , no! In correct!" if the date incorrect is incorrect, or of the date entered in the wrong order. In other words, you compare your input to "July" , "4", "1776" , in that order and you don't case for capital or small case letters.



Problem 6

Create a class **Rounding1**, write the code as shown below. Understand how the rounding methods work. They will be on the quiz soon.

```
12  public class Rounding1
13  {
14          public static void main(String[] args)
15          {
16          System.out.println( "Math.ceil(2.1) returns " + Math.ceil(2.1) );
17          System.out.println( "Math.ceil(2.0) returns " + Math.ceil(2.0) );
18          System.out.println( "Math.ceil(-2.0) returns " + Math.ceil(-2.0));
19          System.out.println( "Math.ceil(-2.1) returns " + Math.ceil(-2.1) );
20          System.out.println( "Math.floor(2.1) returns " + Math.floor(2.1) );
21          System.out.println( "Math.floor(2.0) returns " + Math.floor(2.0) );
22          System.out.println( "Math.floor(-2.0) returns " + Math.floor(-2.0));
23          System.out.println( "Math.floor(-3.0) returns " + Math.floor(-2.1));
24          System.out.println( "Math.rint(2.1) returns " + Math.rint(2.1) );
25          System.out.println( "Math.rint(2.0) returns " + Math.rint(2.0) );
26          System.out.println( "Math.rint(-2.0) returns " + Math.rint(-2.0) );
27          System.out.println( "Math.rint(-2.1) returns " + Math.rint(-2.1) );
28          System.out.println( "Math.rint(2.5) returns " + Math.rint(2.5) );
29          System.out.println( "Math.rint(-2.5) returns " + Math.rint(-2.5) );
            System.out.println( "Math.round(-2.6f) returns " + Math.round(2.6f) );
31          System.out.println( "Math.round(2.0) returns " + Math.round(2.0) );
32          System.out.println( "Math.round(-2.0) returns " + Math.round(-2.0f) );
33          System.out.println( "Math.round(-2.6) returns " + Math.round(-2.6) );
34          }
35  }
```

Problem 7

Create a class **EscapeSequencesAndUnicode**, write the code as shown below. All characters underneath are of size 2 bytes in a sequence of 0s and 1s in UNICODE representation. Understand how the escape characters work from the code below. Unicode character can be entered into a char-variable using exactly-4-digits in hex(line 17). A character can be entered in a character variable as a decimal value (line 16) or as a hexadecimal value (line 16) .They will be on the quiz soon.

```
4   public class EscapeSequencesAndUnicode
5   {
6       public static void main(String[] args)
7           {
8
9           System.out.println( "I delete the last 3 characterssss\b\b\b " );
10          System.out.println( "\t\tI used 2 tabs here " );
11          System.out.println( "I printed a \"  inside quotes " );
12
13          char b = 'b';
14          char bCapital = 66;
15          char a = '\u0061';// 16 * 6 + 1 = 96 in decimal
16          char aCapital = 0x41;
17          char greekAlpha = '\u03b1';// 3 x 256 + 11 x 16 + 1 = ?  in decimal
18
19
20          System.out.println( "I print in latin and in greek ---> "
21                              + b + " "
22                              + bCapital + " "
23                              + aCapital + " "
24                              + a + " "
25                              + greekAlpha );
26          }
27  }
```

Problem 8

Create a class GreekAlphabet. Write EXACTLY 2 lines of code which generate any of the 24 letters of the the Greek alphabet. The first line generates any small case Greek letter. The second line generates any capital Greek letter.

The first small case letter is alpha 'α' and the last letter is omega 'ω'.

The first capital letter is alpha 'Α' and the last letter is omega 'Ω'

Problem 9

Create a class **PasswordMaker1**. Write the code below that generates a password for a student using his initials and age.

```
8   public class PasswordMaker1
9   {
10
11      public static void main(String[] args)
12      {
13        String firstName = "John";
14        String middleName = "Paul";
15        String lastName = "Green";
16        int age = 20;
17          //> extract initials
18        String initials = firstName.substring(0,1) +
19                          middleName.substring(0,1) +
20                          lastName.substring(0,1);
21          //> append age after changing the initials to lower case
22        String password = initials.toLowerCase() + age;
23        System.out.println("Your Password = " + password);
24      }
25  }
```

Problem 10

Create a class **PasswordMaker2**. Write the code which asks the user to enter his first name, last name, and the 4 year digit for the year he/she was born. Read them as String type all 3. Use the method of sub-string of class String to extract the last 2 characters from the first name, the last 2 characters from the last name and the last 2 characters from the year of birth. Concatenate the extractions into a 6 character password and print it.

```
Output - Lab10 (run)
run:
enter first name, last name, year of birth to generate a password for you:
john
green
1994
Your Password = hnen94
BUILD SUCCESSFUL (total time: 21 seconds)
```

Problem 11

Create a class **CharacterClassStringClass**. Ask the user to type 'y' for yes or 'n' for no.
Read the input as String and test if the user type Y\y or N\n which are OK we display the message shown below.
If the user enters any other character except y,Y, N, n display the message shown below.
If the user enters numeric input display the message shown below.
If the user enters a string whose length is greater than 1 we display the message shown below.

Use NESTED IFS AND ELSEs or NESTED IF-ELSEs. No credit for using just IFs without ELSEs

Example code:
String s = "abc";
char c = s.charAt(0) ; // c has the value 'a'