CS 1400- Introduction to Programming and Problem Solving

Team Project # 1
(Due date: April 3rd till midnight)

1.Write a Java program that prompts the user for a month and day and then prints the season determined by the following rules.

Spring	3/21 – 6/20
Summer	6/21 – 9/20
Fall	9/21 – 12/20
Winter	12/21 – 3/20

If an invalid value of month (<1 or >12) or invalid day is input, the program should display an error message and stop. Notice that whether the day is invalid depends on the month! You may assume that the user will never enter anything other than integers (no random strings or floats will be tested.)

Tips:
Break this problem down into smaller pieces. This is always good practice when tackling a larger problem. Break it up into pieces that you can test individually and work on one piece at a time. You might try writing the pseudo-code for each piece.

First, see if you can get a month and ensure that it is valid. Test out only this piece of functionality before continuing. Make sure you test not only the good and bad cases, but also for boundary cases. For example, try entering -5, 0, 1, 4, 12, 13, and 56.

Next, see if you can get a day and ensure that it is valid. Test this piece too.

Finally, use the now-valid month and day to determine which season it is in. If you tested the earlier pieces, you will now know that any bugs are due to a problem here.

2.A palindrome is any word, phrase, or sentence that reads the same forward and backward. The following are some well-known palindromes.

```
Kayak
Desserts I stressed
Able was I ere I saw Elba
```

Create an advanced version of the `PalindromeTester` Program so that spaces, numbers, and punctuations are not considered when determining whether a string is a palindrome. The only characters

considered are roman letters, and case is ignored. Therefore, the `PalindromeTester` program will also recognize the following palindromes:

```
A man, a plan, a canal, Panama
Madam, I'm Adam
Desserts, I stressed
Able was I, ere I saw Elba
Never odd(5,7) or even(4,6)
```

The `PalindromeTester` will continue to run until the user enters a blank line. It will then print out how many palindromes were found. The following are sample interactions that occur when running the program .

Tips:

Remember to break down the problem in pieces. Try to first write the code that inputs one sentence and decides whether it is a regular palindrome where each character counts. Then write the code to ignore spaces, numbers, and punctuations. Finally write the code to loop until the user enters a blank line and count how many palindromes are found.

Test your program thoroughly! Don't test for just the simple cases. If you wanted to break someone else's program, how would you try to do it?

If you can't figure out why your code isn't behaving the way you expected, try printing out intermediate results.

3.Write a program that lets the user play the game of Rock, Paper, Scissors against the computer. The program should work as follows.

When the program begins, a random number in the range of 1 through 3 is generated. If the number is 1, then the computer has chosen rock. If the number is 2, then the computer has chosen paper. If the number is 3, then the computer has chosen scissors. Don't display the computer's choice yet.

The user enters his or her choice of "rock", "paper", or "scissors" at the keyboard. 3. The computer's choice is displayed.

A winner is selected according to the following rules:

- If one player chooses rock and the other player chooses scissors, then rock wins. The rock smashes the scissors.
- If one player chooses scissors and the other player chooses paper, then scissors wins. Scissors cuts paper.
- If one player chooses paper and the other player chooses rock, then paper wins. Paper wraps rock.
- If both players make the same choice, the game must be played again to determine the winner.

Be sure to divide the program into methods that perform each major task. Here is the description of the methods you should implement:

- `public static String computerChoice()` – This method will return the computer's choice of "rock", "paper", or "scissors".

- `public static String userChoice ()` – This method will return the user's choice of "rock", "paper", or "scissors" after validating the user's choice using isValidChoice method.

- `public static boolean isValidChoice (String choice)` – This method will take a String holding the user's choice as parameter and return true if the choice is valid ("rock", "paper", or "scissors"), false otherwise. Be sure that the comparison is not case sensitive.

- public static void determineWinner(string computer,string user)– This method will take both the choices as parameter and display the winner.

Call the method `isValidChoice()` in a `while` loop in the `userChoice()` method to verify that the choice that user enters must be "rock", "paper", or "scissors". If invalid string is input, `isValidChoice()` will return false and the program should ask for new input until the valid input is given.

The user is allowed to play the game as often as desired .

4. Write a `Temperature` class that has two private instance variables:

- `degrees`: a `double` that holds the temperature value

- `scale`: a character either 'C' for Celsius or 'F' for Fahrenheit (either in uppercase or

lowercase)

The class should have
(1) four constructor methods: one for each instance variable (assume zero degrees if no value is specified and assume Celsius if no scale is specified), one with two parameters for the two instance variables, and a no-argument constructor (set to zero degrees Celsius);

(2) two getter methods to return the temperature, one to return the degrees Celsius, the other to return the degrees Fahrenheit – use the following formulas to write the two methods, and round to the nearest tenth of a degree:

```
degreesC = (degreesF - 32) *5 /9
degreesF = (degreesC *9 /5) + 32
```

(3) three setter methods, one to set the value, one to set the scale ('F', 'f', 'C', or 'c', if any other invalid character is given, terminate your program), and one to set both;

(4) three comparison methods, an equals method to test whether two temperatures are equal, one method to test whether one temperature is greater than another, and one method to test whether one temperature is less than another (note that a Celsius temperature can be equal to a Fahrenheit temperature as indicated by the above formulas).

Write a driver program in `TemperatureTest` that tests each of the constructors, getters, setters, and include at least one true and one false case for each of the comparison methods.

Note: Each member will submit me the project individually on the blackboard to be scored . Make sure to mention under comment your name and your team mate name on every question of the project.

For Example:

//John Shake

//Richard daye

//CS 1400-Section

//project 1