

school of **computing, informatics,
& decision systems engineering**

CSE 110 – Assignment #4

Maximum points: 20 pts

Topics

- Object Oriented Programming (Chapter 8)
 - Encapsulation
 - Implementing classes
 - Implementing methods
 - Object construction
 - Constructors

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. **(If not, -1 Pt)**

```
//*****  
// Name: your name  
// Title: title of the source file  
// Description: Write the description in your words.  
// Date: the date you programmed  
//*****
```

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 <https://engineering.asu.edu/integrity/>

Part 1: Test Program Development (5 pts)

Step 1 (2 Pts): Write a Java program called **Assignment4.java**, which reads the user input command and display a comment corresponding to the command.

- Command A: Display " *** Make A new FiveCards *** "
- Command B: Display " *** Change One Card ***"
- Command C: Display " *** Display Data ***"
- Command Q: Display " *** End of Program *** ", and quit the program
- Others : Display " *** Invalid command. Try Again *** "

The following is the CommandTemplate.java used in the previous project. Start from modifying this code if you have no idea what to do.

```
1 import java.util.Scanner;
2 public class CommandTemplate {
3     public static void main (String[] arg){
4         char command;
5         Scanner in = new Scanner(System.in); // user input scan
6         int num0 = 0;
7         int num1 = 0;
8         do {
9             System.out.println("Choose (A: add data), (R: refresh data), (D: Display data) or (Q: quit)");
10            String temp = in.nextLine();
11            command = temp.toUpperCase().charAt(0);
12
13            System.out.println(command);
14            if (command == 'A'){
15                System.out.println("Input Qs and Ia in one line");
16            }
17            else if (command == 'R'){
18                System.out.println("The data is refreshed");
19            }
20            else if (command == 'Q'){
21                System.out.println("*** End of Program ***");
22            }
23            else {
24                System.out.println("Invalid command. Try Again");
25            }
26            System.out.println();
27        } while (command != 'Q');
28    } // end of main
29 } // end of class
```

Step 2 (3 Pts): Go to the Part2 and develop **FiveCards.java** in the same directory of Assignment4.java. Once the Part2 is finished, modify each command block in Assignment4.java as:

- The top level: Delete the num0 and num1 if you have them, and add the statement at the line after Scanner initialization.

```
FiveCards myCards = new FiveCards();
```

- Command A: After displaying the message, add the statements:


```
System.out.println("Type five letters without space");
String str = in.nextLine();
myCards.setCard(str);
myCards.calculateScore();
System.out.println(myCards.displayData());
```
- Command B: After displaying the message, add the statements:


```
System.out.println("Type one position to change");
int pos = in.nextInt();
String lineBreak = in.nextLine(); //to skip the line break
myCards.changeOne(pos);
myCards.calculateScore();
System.out.println(myCards.displayData());
```
- Command C: After displaying the message, add the statement:


```
System.out.println(myCards.displayData());
```

Part 2: Programming (15 pts)

This assignment is to write a class definition (not a program, no main method) named **FiveCards** (saved in a file **FiveCards.java**). The class has the following instance variables:

```
private String cards
private int score
private int changes
private String history
```

- The **cards** is a string data of five characters among "1234567890JQK". No space between the five letters.
- The **score** is 0 for no pair, 1 for one pair, 2 for two pairs, 3 for three cards, and 4 for full-house (one pair and three cards).
- The **changes** is a counter to keep the number of calling change functions.
- The history is a string data to keep all history of five cards such as following lines.

```
cards      score      0 \n
cards      score      1 \n
...
```

The FiveCards class must include the following constructor and methods: (If your class does not contain any of the following methods, points will be deducted).

Method	Description of the Method
public FiveCards ()	Make a new FiveCards object and initialize the instance variables cards, score, changes, and history as "", 0, 0, and "" (2 Pts).
public void setCards (String str)	Set the cards with the input string data. (2 Pts).
public void changeOne (int pos)	Replace one character of the given position [0,4] with a random letter from "1234567890JQK". Increase the value of changes by one. (3 Pts)
public void calculateScore ()	Calculate the pairs and update the score as: 0: No pair among 5 letters in the cards . 1: One pair in the cards 2: Two pairs in the cards 3: Three same letters in the cards 4: Full House (one pair and three cards) You may use the calculatePair (char) 13 times for each letter, which is explained below. (5 Pt)
private int calculatePair (char c)	Return how many times the input character is used in the cards. For example, when the cards is "01KQK", the calculatePair('K') returns 2. *) This may be NOT necessary, if you do in the different way.
public String displayData ()	Call the calculateScore () in the method to update the score, and return a string data of history for each change. Don't use System.out method in the method. Look at the format in Sample Output (3 Pts)

Sample Score and Change

[Cards]	[Score]	[Changes]
12345	0	1
22345	1	2
22 335	2	3
222 35	3	4
22255	4	5
72255	2	6
722 59	1	7
7K259	0	8

Tips:

From this assignment, it will be asked to submit multiple files. One of them **does not compile, ZERO (0) point may be given**. To avoid this tragedy and get some partial points, make two files that compile successfully first even though it is not completed.

Start from DUMMY methods. The following are the examples of Dummy method. For the void type method, keep the body empty. For the return-type method, add one statement to return dummy data of required data type. There is no error, so it compile successfully. In this assignment, make 5 dummy methods first. If the FiveCards has only dummies, you are guaranteed to get 5 points of Part#1, if it does compile.

Then start to complete the method one by one and test one by one. Do not implement all methods at the same time. It will be nightmare to fix many items together.

```
void XXX(int x) {      }
String YYY(int x, int y){return "test";}
```

Note:

- Make only one `System.in`. If your program does not run correctly because of this issue, you lose the points **(-3Pts)**.
- Use only the Java statements that have been covered in class to date. **DO NOT use any other items out of the Chapter 1- 5, and 8 (Array, Array List, exit() etc are not allowed to use)**. If in doubt, ask the instructor or TA. If you use them, then you lose the points of task. **(-3Pts)**.

Example Executions:

The following are example inputs and outputs. The user inputs are shown **in red, which will not be displayed when you submitted online**. The result may be different from this because the program generates **random** letters.

Program Input (1)

```
-----
INPUT 1
-----
A
12345
A
22345
A
22335
A
22235
A
22255
B
0
B
4
B
1
Q
-----
```

YOUR OUTPUT 1

```
-----
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
A
*** Make A new FiveCards ***
Type five letters without space 12345
[Cards][Score][Changes]
12345 0 1
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
A
*** Make A new FiveCards ***
Type five letters without space 22345
[Cards][Score][Changes]
12345 0 1
22345 1 2
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
A
*** Make A new FiveCards ***
Type five letters without space 22335
[Cards][Score][Changes]
12345 0 1
22345 1 2
22335 2 3
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
A
*** Make A new FiveCards ***
Type five letters without space 22235
[Cards][Score][Changes]
12345 0 1
22345 1 2
22335 2 3
22235 3 4
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
A
*** Make A new FiveCards ***
Type five letters without space 22255
[Cards][Score][Changes]
12345 0 1
22345 1 2
22335 2 3
22235 3 4
22255 4 5
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
B
*** Change One Card ***
Type one position to change 0
[Cards][Score][Changes]
12345 0 1
22345 1 2
22335 2 3
22235 3 4
22255 4 5
92255 2 6
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
B
*** Change One Card ***
Type one position to change 4
[Cards][Score][Changes]
12345 0 1
```

```

22345      1      2
22335      2      3
22235      3      4
22255      4      5
92255      2      6
92254      1      7
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
B
*** Change One Card ***
Type one position to change 1
[Cards][Score][Changes]
12345      0      1
22345      1      2
22335      2      3
22235      3      4
22255      4      5
92255      2      6
92254      1      7
97254      0      8
Choose (A: new), (B: changeOne), (C: Display), or (Q: quit)
Q
*** End of Program ***

```

/*****
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 two files **Assignment4.java** and **FiveCards.java** on-line in Assignment#4 item in Gradescope at the same time. (Drag and Drop two files together)
- In comments of each file, the assignment Header including your name, title, lab, date, etc.
- 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.**