Computer Engineering Department Kuwait University

**CpE 201/01**

**Object Oriented Paradigm**

**HW#4**

**Due: 18th October 2017 at 23:55**

**Important Notes:**

1. **All files should be in one folder hw4YOURNAME (hw4Sara)**
2. **All files should be .java files (q1sara.java, q2sara.java, ...) with your Name & ID as comment in the top of the programs**
3. **Don't forget to change class name to match file name (q1sara.java --> class name should be q1sara)**
4. **Compress your folder and upload**
5. **Don't wait for the last minute, the upload my take sometime and if the submission is closed it will not be reopened**
6. **The files should also contain the output as a comment at the end of the program between /\*\*/**
7. **You have two jokers to be used for the homework. Each joker will give you a 24 hours delay. Once the homework is discussed in class no joker is accepted**
8. Write a program that reads a number from the user and print it in backward and forward direction with spaces. Input a number of any size (long int). Print each digit of the number in backward direction with spaces between the digits. Print the digits in Forward direction with spaces between them. Use while loops

**Enter a number: 34562**

**Backward**

**2 6 5 4 3**

**Forward**

**3 4 5 6 2**

1. Write an application that inputs a 3-digits number from the user and checks if all digits are even numbers. If a number is entered with an odd digit your program should stop. Use **do/while** for reading the input and any loop format to test if the number is all even or not.

**Enter a 3-digit number: 224**

**All even!**

**Enter a 3-digit number: 642**

**All even!**

**Enter a 3-digit number: 252**

**An odd digit was found (5)!**

**Thank You!**

1. Modify the above program by creating class ***Evens*** that has the following instance variables:
   * *Int* ***digit1****: That stores the first digit*
   * *Int* ***digit2****: That stores the second digit*
   * *Int* ***digit3****: That stores the third digit*

Your class should also have the following methods:

* *A constructor* ***Evens*** *that takes a three digit number (Evens (int num);)*
* *A method* ***parser*** *that is used by the constructor to parse the number into the 3 digits (void parser (int num);)*
* *Three* ***set*** *methods that are used by the constructor to set the three instance variables (void setDigit1(int x); void setDigit2(int y);void setDigit2(int z); )*
* *Three* ***get*** *methods that are used to retrieve the three instance variables (int getDigit1(); int getDigit2();int getDigit2(); )*
* *A method checker that is used to check if all the digits are even or not. This method will either print (All Even!) or will print the odd digit. This method will also return false if an odd digit exist and true if all digits are even (Boolean checker();)*

Write a test application to demonstrate the class *Evens*.

**Enter a 3-digit number: 224**

**All even!**

**Enter a 3-digit number: 642**

**All even!**

**Enter a 3-digit number: 252**

**An odd digit was found (5)!**

**Thank You!**

1. Write a program that reads two lines equations from the user and show if the lines intersect or not. To find whether two points intersects or not use the following steps:
2. Set the two equations for y equal to each other.
3. Solve for x. This will be the x-coordinate for the point of intersection. If the x is undefined (denominator of zero) this will mean that the two points doesn’t intersect.
4. Use this x-coordinate and plug it into either of the original equations for the lines and solve for y. This will be the y-coordinate of the point of intersection.
5. As a check for your work plug the x-coordinate into the other equation and you should get the same y-coordinate.

You now have the x-coordinate and y-coordinate for the point of intersection. You need to draw the two lines by choosing two points on the lines and drawing the lines using GUI.



**Enter first equation y = a1x+b1 (a1, b1): 20 30**

**Enter second equation y = a2x+b2 (a2, b2): 30 70**

**The lines intersect in the point (-4, -50)**

**Enter the first point on the first line x coordinate: 0**

**The point is (0, 30)**

**Enter the second point on the first line x coordinate: 10**

**The point is (20, 230)**

**Enter the first point on the second line x coordinate: 0**

**The point is (0,70)**

**Enter the second point on the second line x coordinate: 30**

**The point is (30,970)**