Final Project -Spring Semester 2019
CS 1400-02
Due date of submission: May 16th Till Midnight
Total Points:100

1.Design a class named Employee. The class should keep the following information in fields:
- Employee name
- Employee number in the format XXX-L, where each X is a digit within the range 0-9 and the L is a letter within the range A-Z.
- Hire date

Write a private method to validate input employee number. If it is not valid, assign empty string to the corresponding instance field. Write a no-arg constructor that sets all fields to empty string, a 3-arg constructor, appropriate setter and getter methods, and the toString method. The toString method should concatenate "INVALID EMPLOYEE NUMBER" if the employee number is invalid.

(b) Write a class named ProductionWorker that extends the Employee class. The ProductionWorker class should have fields to hold the following information:
- Shift (an integer)
- Hourly pay rate (a double)

The workday is divided into two shifts: day and night. The shift field will be an integer value representing the shift that the employee works. The day shift is shift 1 and the night shift is shift 2. Write a no-arg constructor that sets the default shift to 1 and pay rate to 0.0, a 5-arg constructor, appropriate setter and getter methods, and the toString method for the class. The toString method should concatenate "INV ALID SHIFT NUMBER" if the shift number is neither 1 nor 2. Print hourly pay rate with $ in the front and two digits after decimal point.

(c) Write a driver program WorkerTest to create ProductionWorker objects and test all features and capabilities described above. For example, create the first ProductionWorker object and pass the initialization data to the constructor. Now create another ProductionWorker object and use the default constructor and set methods. Finally, create a third ProductionWorker object with wrong employee number and invalid shift integer. The toString methods should generate the following output:

```
$ java WorkerTest Here's the first production worker.

Name: John Smith
Employee Number: 123-A
Hire Date: 11-15-2005
Shift: Day
Hourly Pay Rate: $23.50

Here's the second production worker.
```

Name: Joan Jones
Employee Number: 222-L
Hire Date: 12-12-2018
Shift: Night
Hourly Pay Rate: $18.50
Here's the third production worker.
Name: Tony Gaddis
Employee Number: INVALID EMPLOYEE NUMBER
Hire Date: 1-23-2006
Shift: INVALID SHIFT NUMBER
Hourly Pay Rate: $19.50

2.Write a class Fraction for representing fractions and performing arithmetic with them. The Fraction class has the following private fields:
numerator: an integer
denominator: an integer
created: a static integer to count the number of fractions that are created so far

The class should have the following public methods:

constructor: A noarg constructor that sets the default fraction to 0/1 constructor: Accepts a numerator and a denominator as arguments
add: performs addition of two fractions and returns a new fraction
subtract: performs subtraction of two fractions and returns a new fraction multiply: performs multiplication of two fractions and returns a new fraction divide: performs division of two fractions and returns a new fraction toString: returns a string for the fraction in a form like 2/3 printAsFloat: prints the fraction in a form like 0.66667 numberOfFractions: A static method to return the number of fractions which have been created so far
read: a static method to prompt the user to enter the numerator and denominator of a fraction; the method returns the created fraction.
Your constructor should take care of the following cases:
- Prevent creation of fractions with 0 in the denominator. Print a warning message and change the fraction to 0/1.
- Negative signs, if any, should appear only in the numerator. Thus, -1/2 is acceptable, but 1/-2 is not and should be changed to -1/2.
- Anything with 0 in the numerator and non-zero denominator — *e.g.* 0/-1 or 0/2 — should reduce to 0/1.
- The numerator and denominator must not have any common divisors. The following code computes the greatest common divisor (gcd) of two integers. In order to reduce a fraction to lowest terms, you should divide both the numerator and denominator by their gcd.

// computes and returns the gcd of the two positive parameters // by the Euclid's algorithm

```
private int gcd (int num1, int num2)
{
  while (num1 != num2)
  {
    if (num1 > num2)
      num1 = num1 - num2;
    else
      num2 = num2 - num1;
}
  return num1;
}
```

Write a separate private method to normalize data as described above, and have this method be called by the regular constructor.
Example. Driver code like this —

```
   Fraction c = new Fraction(1,3);
   Fraction d = new Fraction(2,4);
   Fraction e = new Fraction();
   System.out.println("e = " + e);
e = c.add(d);
System.out.print(c + " + " + d + " = " + e + " = "); e.printAsFloat();
System.out.println();
...
```

would produce output like this —

```
   e = 0/1
   1/3 + 1/2 = 5/6 = 0.8333333
   ...
```

Write a driver program FractionTest.java to test your class. You must demonstrate all of the capabilities and features described above. The following are sample interactions when running the program:

```
$ java FractionTest Please enter 2 fractions --
Fraction 1:
enter an integer numerator: 1
enter an integer denominator: 2 Fraction 2:
enter an integer numerator: 1 enter an integer denominator: 3 1/2 + 1/3 = 5/6 = 0.8333333
1/2 - 1/3 = 1/6 = 0.16666667
1/2 * 1/3 = 1/6 = 0.16666667
1/2 / 1/3 = 3/2 = 1.5
6 fractions have been created.
```

```
$ java FractionTest Please enter 2 fractions --
Fraction 1:
enter an integer numerator: 3
enter an integer denominator: 0 denominator cannot be 0
the fraction is set to 0/1 Fraction 2:
enter an integer numerator: 6 enter an integer denominator: -2 0/1 + -3/1 = -3/1 = -3.0
0/1 - -3/1 = 3/1 = 3.0
0/1 * -3/1 = 0/1 = 0.0
0/1 / -3/1 = 0/1 = 0.0
6 fractions have been created.
```

3.The Department of Motor Vehicles has asked you to write a program that grades the written portion of the driver's license exam. The exam has 20 multiple choice questions. A student must correctly answer 15 of the 20 questions to pass the exam. Here are the correct answers:

```
 1. A     6. B    11. C    16. D
 2. A     7. B    12. C    17. D
 3. A     8. B    13. C    18. D
 4. A     9. B    14. C    19. D
 5. A    10. B    15. C    20. D
```

Write the following methods in the program:

• public static int gradeExam(char[] correct, char[] student) – This method will determine the number of incorrect answers. Be sure that the comparison is not case sensitive.
public static int[] makeMissedArray(char[] correct, char[] student, int numIncorrect)- This method will make the missed array and stores the numbers of all the questions that the student missed in it.

• Write a main method that holds the correct answers, asks the user to enter a student's answers, call the above two methods, and then displays the results. The program should use a constant for the maximum number of multiple choice questions. The following are examples of the **required** I/O behavior, where the user's input is shown in bold.

```
project $ java DriverTest Enter your answers to the exam questions. Question 1: A
Question 2: A
Question 3: A
Question 4: A
Question 5: A
Question 6: B
Question 7: B
Question 8: B
Question 9: B
Question 10: B
Question 11: C
Question 12: C
Question 13: C
Question 14: C
Question 15: C
Question 16: A
Question 17: A
Question 18: A
Question 19: A
Question 20: A
** Correct answers: 15
** Incorrect answers: 5
** You passed the exam.
You missed the following questions: 16 17 18 19 20

$ java DriverTest Enter your answers to the exam questions.
Question 1: q
Question 2: q
Question 3: q
Question 4: a
Question 5: a
Question 6: b
Question 7: b
Question 8: b
Question 9: b
Question 10: q
Question 11: c
Question 12: q
Question 13: c
Question 14: c
Question 15: c
Question 16: q
Question 17: d
Question 18: d
Question 19: d
Question 20: d
** Correct answers: 14
** Incorrect answers: 6
** You failed the exam.
You missed the following questions: 1 2 3 10 12 16
```

3.Write a program that produces simple graphics on a screen by placing ordinary keyboard characters at certain places on each line.

You will create two classes, one for a box and one for a triangle. Each of the figures, the box and the triangle, will have an offset telling how far they are indented from the edge of the screen. Each figure will also have a size, although the size will be determined differently for a box and a triangle. For a box, the size is given as the width and the height, expressed in the number of characters. Because characters are taller than they are wide, a box will look taller than you might expect when it is written on the screen. For a triangle, the size will be given by its base. The triangle will point up, and the base will be at the bottom. The slope of a side of the triangle is limited to what you get by indenting one character per line. So once the base is chose, you have no choice regarding what the sides of the triangle will be. We also assume that the base size of a triangle must be odd. The following shows a sample of a 5-by-5 box and a triangle with base of size 15. Assuming that the offset 10.

```
          -----
          |   |
          |   |
          |   |
          -----

                 *
                * *
               *   *
              *     *
             *       *
            *         *
           *           *
          ***************
```

The following are the figures of the same size but with the offset 20.

```
-----
|   |
|   |
|   |
-----

       *
      * *
     *   *
    *     *
   *       *
  *         *
 *           *
***************
```

The base class Figure will have instance variables for any properties that all figures have in common and will have methods for the actions that all figures have. For example, every figure has a given offset, and every figure should be able to draw itself. However, because we don’t know how to draw a figure of unknown shape, declare drawHere() to be an abstract method and Figure to be an abstract class.
The classes Box and Triangle will then be derived classes of Figure and they will provide the implementation of drawHere(). Here is the UML diagram of Figure, Box, and Triangle.

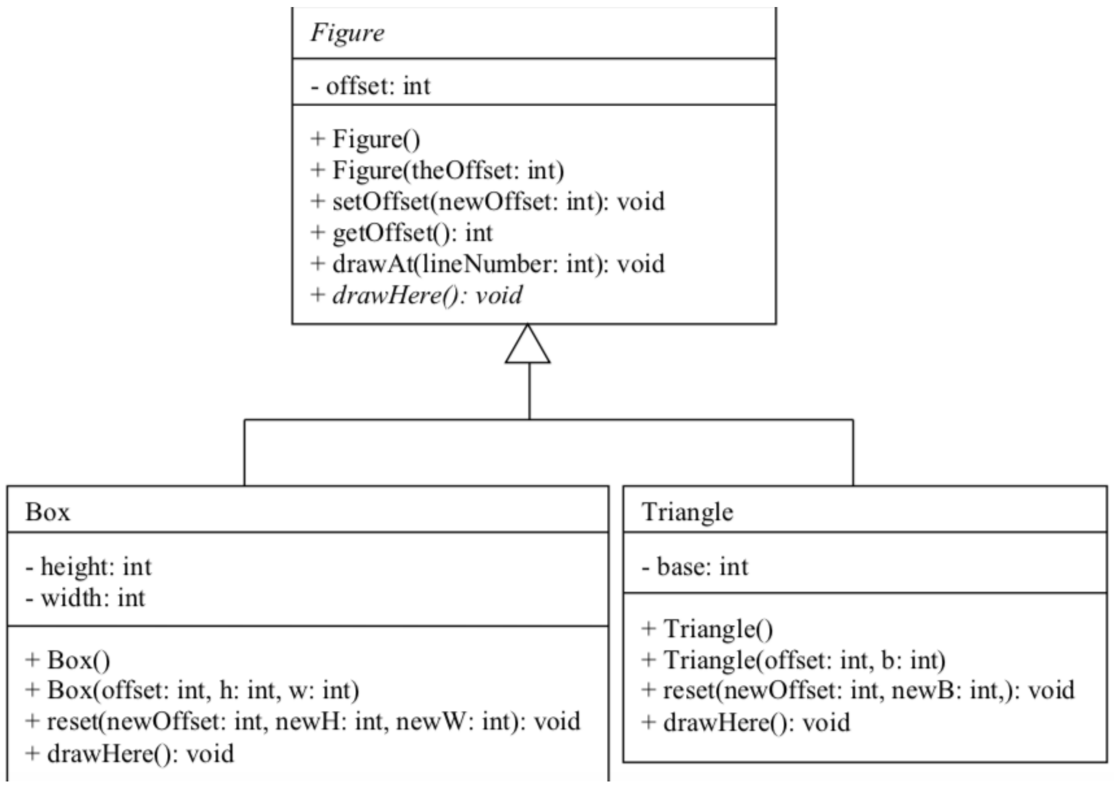
Figure
- offset: int
+ Figure()
+ Figure(theOffset: int)
+ setOffset(newOffset: int): void
+ getOffset(): int
+ drawAt(lineNumber: int): void
+ drawHere(): void
Box
- height: int
- width: int
+ Box()
+ Box(offset: int, h: int, w: int)
+ reset(newOffset: int, newH: int, newW: int): void
+ drawHere(): void
Triangle
- base: int
+ Triangle()
+ Triangle(offset: int, b: int)
+ reset(newOffset: int, newB: int,): void
+ drawHere(): void

The method drawAt has one parameter of type int. The method drawAt inserts a number of blank lines equal to this parameter and then draws the figure by calling drawHere. When you override drawHere, then drawAt will also produce correct figures. Here is the implementation of drawAt.

```
public void drawAt(int lineNumber)
{
  for (int count = 0; count < lineNumber; count++)
    System.out.println();
  drawHere();
}
```

Write a program to test your classes. You must demonstrate all of the capabilities and features described above. Draw some interesting figures including a Christmas tree as follows:

```
          *
         * *
        *   *
       *     *
      *       *
     *         *
    *           *
   *             *
  *               *
 *                 *
*********************
        -----
        |   |
        |   |
        -----
```

Here is the code in main of the test driver to draw the above Christmas tree:
Triangle top = new Triangle (5, 21);
Box base = new Box(13, 4, 5);
Top.drawAt(1);
Base.drawAt(0);

Note:
All the program should be submitted in .java Extension Files in the blackboard by every individual with the screenshot of all the 4 programs

There should be a following comment mentioned on the top of your project:

//Final Project
// Team Member name A,B
//Date
//Class,Section

Also If somebody is working individually they won't get extra credit points . It's their choice and responsibility to complete it on time.

All The best

Thankyou

Devika Maini

Email Me at dkakkar@cpp.edu for any further question