Assignment 01 – Pizza Party

You must work in **alone** on this assignment. Do not use any Java language features we have not cover so far in this course.

Assignment Objectives

After completing this assignment the student should be able to:

- Write, compile, and run a small Java program
- Use **System.out** to display prompts and other information to the user
- Collect input using a **Scanner** object
- Declare and use variables
- Use math operators and expressions to compute desired values

Assignment Requirements

For this assignment you are given the following files:

Assignment01.java (you must complete this file)

Problem Description and Given Info

Within the **main** method in the **Assignment01.java** file, you must write a program to determine how to split a number of pizzas (each with the same number of slices) among a group of adults and children at a pizza party.

Your program must prompt the user to enter the 4 integer input values described below. It must collect the user's input and store these values in 4 different variables. It must collect the inputs in the order shown below.

Your program must compute the 5 integer output values described below. It must store these output values in 5 different variables. It must print out the 5 output values. The output must be formatted exactly like the Expected Outputs examples shown below.

Inputs

1. Number of pizzas purchased
2. Number of slices per pizza
3. Number of adults
4. Number of children

Outputs

1. Total number of slices of pizza
2. Total number of slices that will be given to the adults
3. Total number of slices available for the children
4. Number of slices each child will get
5. Number of slices left over

Other Details

- Each adult will get two slices of pizza
- Remaining slices will be divided evenly among the children
- Each child will get the same number of slices of pizza
- No pizza slices will be split or divided
- Make sure that your program displays 5 lines of output that look like these:

```
Total number of slices of pizza              : 8
Total number of slices given to adults       : 2
Total number of slices available for children : 6
Number of slices each child will get         : 6
Number of slices left over                   : 0
```

Test Data

Test #1

Given Inputs

```
Number of pizzas purchased : 1
Number of slices per pizza : 8
Number of adults           : 1
Number of children         : 1
```

Expected Outputs

```
Total number of slices of pizza               : 8
Total number of slices given to adults        : 2
Total number of slices available for children : 6
Number of slices each child will get          : 6
Number of slices left over                    : 0
```

Test #2

Given Inputs

```
Number of pizzas purchased : 2
Number of slices per pizza : 8
Number of adults           : 2
Number of children         : 3
```

Expected Outputs

```
Total number of slices of pizza               : 16
Total number of slices given to adults        : 4
Total number of slices available for children : 12
Number of slices each child will get          : 4
Number of slices left over                    : 0
```

Test #3

Given Inputs

```
Number of pizzas purchased : 2
Number of slices per pizza : 10
Number of adults           : 3
Number of children         : 8
```

Expected Outputs

```
Total number of slices of pizza               : 20
Total number of slices given to adults        : 6
Total number of slices available for children : 14
Number of slices each child will get          : 1
Number of slices left over                    : 6
```

What to turn in

For this assignment you must upload the following file(s) by the due date.

Assignment01.java

Any assignment submitted **less than 24 hours after the posted due date** will have **10 points deducted**.

Any assignment submitted **more than 24 hour after the posted due date** will receive **a zero in the grade book**.

Grading Rubric

Criteria	Points
All required files are submitted	10
Each file includes a comment header with the following information: • `CSE 110 : <Class #> / <meeting days and times>` • `Assignment : <assignment #>` • `Author : <name> & <studentID>` • `Description : <of the file contents>` • Partial credit can be awarded	
Code is neat and well organized	10
• Good naming conventions for all identifiers • Good use of whitespace • Descriptive comments • Partial credit can be awarded	
Code compiles with no syntax errors	20
• No Partial credit can be awarded • No credit will be awarded for structure or logic if your code does not compile	
Code passes structure tests	30
• Code collects 4 required inputs • Code computes and stores 5 values • Code outputs 5 results • Partial credit can be awarded	(10) (10) (10)
Code passes logic tests	30
• Partial credit is awarded based on number of tests passed • No credit will be awarded for logic if your code does not pass all structure tests • See test examples (#1 - #3) above in these instructions	
TOTAL	**100**