

## PROG 1100 - 702 - Assignment 2

### Decisions and Procedures

74

Assignment Value: 13% of overall course mark.

Due Date: Thursday, October 10<sup>th</sup> by 10:30AM (ie. *Before* class begins)

Late submissions will receive the standard late submission penalty as stated in the course outline. (10% overall deduction per 12-hour block late)

Use VB.NET to create a program for the following problems. You will maintain and store your code using your Team Foundation Services (TFS) account that you created in your first class. Your code in TFS will be accessible by your instructor for review.

Use a separate Visual Studio solution for each of the questions in this assignment.  
Use the following naming convention for your solutions.

[Lastname]\_[Firstname]\_Assignment2A  
[Lastname]\_[Firstname]\_Assignment2B  
[Lastname]\_[Firstname]\_Assignment2C

#### A) Restaurant Menu (PROGRAMMING PROJECT 5 on page 162 of your textbook)

Write a program to place an order from the following menu...

Burgers	Fries	Drinks
Regular (4.19)	Small (2.39)	Soda (1.69)
w/ cheese (4.79)	Medium (3.09)	Bottled Water (1.49)
w/ bacon (4.79)	Large (4.99)	
w/ bacon and cheese (5.39)		

Write the program so that each group box is invisible and only becomes visible when its corresponding check box is checked. After the button is clicked, the cost of the meal should be calculated. (Note: the Checked property of the first radio button in each group should be set to True in its Properties window. This guarantees that a selection is made in each visible group box. Of course, when the cost of the meal is calculated, only the visible group boxes should be considered.)

*Refer to Figures 4.62 and 4.63 on page 163 in your textbook for a visual representation of the outcome of this program.*

### B) Grade Point Average (PROGRAMMING PROJECT 1 on page 218 of your textbook)

Write a program to calculate a student's GPA. The user should enter the grade (A, B, C, D, or F) and the number of credit hours for a course and click on the *Record This Course* button. The user should then repeat this process for all his or her courses. After all the courses have been recorded, the user should click the *Calculate GPA* button. A function procedure should be used to calculate the quality points for a course.

Use the following formula for calculating the quality points for the course:

~~The program must employ a function named *CalculateQualityPoints* which returns a decimal number between 0 and 4 based on whether the user entered a grade of A, B, C, D or F combined with the number of credits the user entered.~~

~~—A = 4, B=3, C=2, D=1, and anything else should = 0~~

~~—The function should calculate the quality points for a course as the numerical grade multiplied by the number of credits that the course is worth.~~

#### CORRECTION:

The program must employ a function named *CalculateQualityPoints* which returns an integer based on the following formula.

$$\text{qualityPoints} = \text{numericalGrade} * \text{numberOfCreditHours}$$

*(You can reference figure 5.38 in your textbook for a visual representation of the outcome of this program)*

### C) Yahtzee! Game (Not in your textbook)

Write a simple variation of the popular dice game Yahtzee which does the following:

- The user will click a button on a form application which will simulate the rolling of five separate dice.

- After the dice have been rolled, it should be determined by the program what the highest combination of dice was rolled

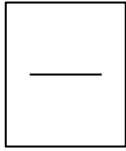
- For example, if the values of the dice rolled were 2 2 5 5 6, then it should be determined by the program that the highest combination in that specific dice roll was a pair. (Either the pair of twos or the pair of fives)

- If the values rolled are 4 5 6 4 4, then the program should identify a combination of three-of-a-kind in that specific roll.

- If the values rolled are 1 2 3 4 5 then the program should identify that no matches or combinations were found in that specific dice roll.

- The same would apply for four-of-a-kind and five-of-a-kind (aka Yahtzee!)

- Every time the dice are rolled and the determination of combinations is made, the results should be stored in a running count of the following statistics. These statistics should be displayed to the end user as a running total:



- The number of total dice rolls
  - The number of rolls without any combinations.
  - The number of two-of-a-kind rolls
  - The number of three-of-a-kind rolls
  - The number of four-of-a-kind rolls
  - The number of Yahtzee! rolls
  - The percentage of rolls without any combinations to two decimal points.
  - The percentage of two-of-a-kind rolls to two decimal points.
  - The percentage of three-of-a-kind rolls to two decimal points.
  - The percentage of four-of-a-kind rolls to two decimal points.
  - The percentage of Yahtzee! rolls to two decimal points.
- The user should also be able to click a Reset button at any time and clear out the dice and the statistics should all be reset to zero.
- If the user happens to roll five-of-a-kind (Yahtzee!), then the program should show a message box with the word YAHTZEE! to emulate the shouting of the word in the actual game. (See screenshot below)

### Requirements:

- Because of the random nature of the dice roll game, you will have to use a built in VB object called *Random*. This random object will need to simulate a dice roll by generating a number between 1 and 6 (five times per roll)  
You should research in your book how to generate these numbers.
- You will be provided images (download from Moodle) of each side of a die which you will need to include in your application. These images should be stored as *Resources* within your application.
- Because the purpose of this assignment is to explore and master the concept of Subs and Functions, you are required to use the following procedures in your code and implement the functionality for each of the following procedures that will accomplish these tasks...

### Processing

```
Sub UpdateStats(highestMatch As Integer,  
    ByRef numberOfRolls As Integer,  
    ByRef numberOfNoMatches As Integer,  
    ByRef numberOfPairs As Integer,  
    ByRef numberOfThreeOfAKind As Integer,  
    ByRef numberOfFourOfAKind As Integer,  
    ByRef numberOfYahtzees As Integer,  
    ByRef noMatchesPercentage As Double,  
    ByRef pairsPercentage As Double,  
    ByRef threesPercentage As Double,  
    ByRef foursPercentage As Double,
```

```
ByRef yahtzeePercentage As Double)
```

```
Function GetHighestCombination(firstDiceValue As Integer,  
                                secondDiceValue As Integer,  
                                thirdDiceValue As Integer,  
                                fourthDiceValue As Integer,  
                                fifthDiceValue As Integer) As Integer
```

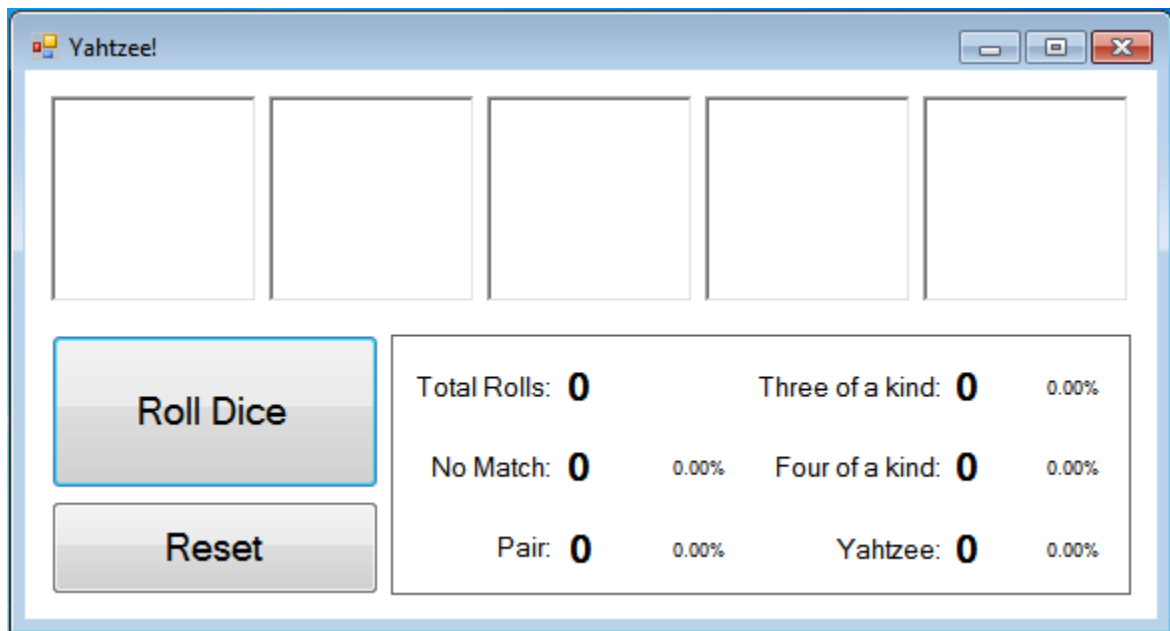
### Output

```
Sub DisplayDiceRoll(firstDiceValue As Integer,  
                    secondDiceValue As Integer,  
                    thirdDiceValue As Integer,  
                    fourthDiceValue As Integer,  
                    fifthDiceValue As Integer)
```

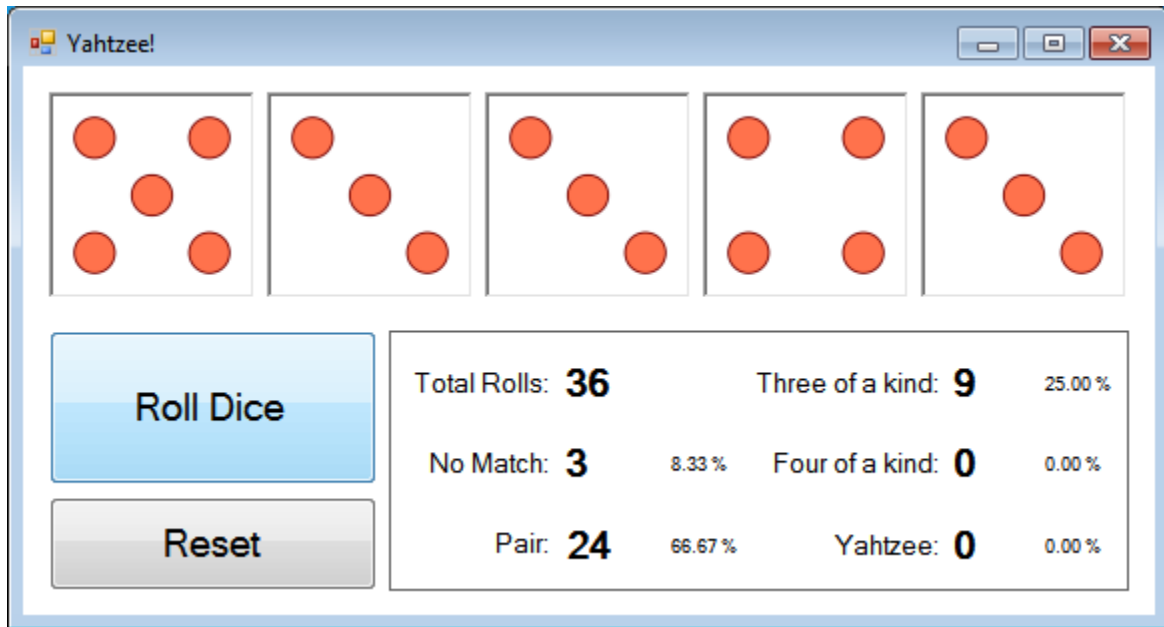
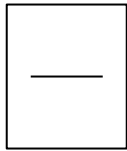
```
Sub DisplayStats()
```

### Sample Screenshots

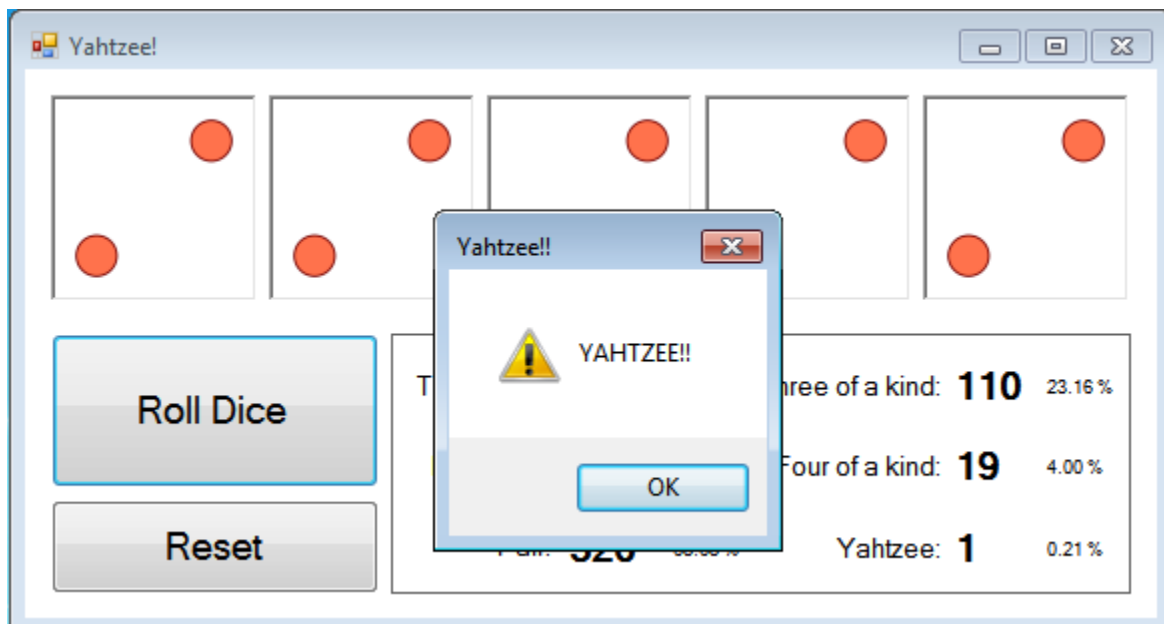
At Design/Startup...



After a few rolls of the dice...



Yahtzee!!



Bonus additions (not required but will reward bonus marks):

- Handle and display the tracking for these additional combinations of dice
  - A full house (three of a kind and a pair) eg. 5 5 5 3 3
  - A straight of five dice eg. 1 2 3 4 5

## Grade Calculation

Question #A: \_\_\_\_\_ / 14

Question #B: \_\_\_\_\_ / 20

Question #C: \_\_\_\_\_ / 40

**Total: \_\_\_\_\_ / 74**