

Final Exam is Monday May 1, 2017 at 8:00am

1. Write a method named *EvenOdd* which will accept a string and a bool as arguments. If the bool is true the method should return a new string made of the even characters of the string argument. Otherwise, it should return the a new string made of the odd characters of the string argument.

2. Write a method which accepts a string argument and returns a string result. The result string will be made up of only the original string words – all spaces, and punctuation should be removed.

3. Write a method which will accept a string and return the number of three-letter words in the string.

4. Write a method which will receive two string arguments and return an int. The function should do the following:

- If the two strings are not the same length it should return -1.
- If the two string match exactly, it should return the length of the string.
- Otherwise, it should return the index of the first string position where the two strings do not match.

5. A *Bananas* is defined below. Answer the following questions:

A) Show how to declare a variable of type Bananas:

B) What are the private variable in the class?

C) Show how the third overloaded constructor might be used in a main program.

```
{class Bananas
{public Bananas()
  {a = 0;
  b = 0;
  }
public Bananas(int x)
  {a = x;
  b = 0;
  }
public Bananas(int x, int y)
  {a = x;
  b = y;
  }
private int a, b;
}
```

6. Write a C# program to input a character from the keyboard. If the character is 'A' run MethodA() and input another character from the keyboard. Do the same for the characters 'B' and 'C' except run MethodB() and MethodC() respectively. If the character is a 'D' terminate the program. Ignore all other characters. To get full credit for this program you must use a switch structure.

7. The following method does a Bubble sort on a parameter named sortVector. Line numbers have been added to the left for reference purposes. Answer the questions below with regard to this method.

```
1 private void Sort(int[] a)
2   {bool sorted = false;
3     int i, j, k, iTmp;
4     while(sorted == false)
5       {sorted = true;
6         for(i=0;i<a.length - 1;i++)
7           {j = a[i];
8             k = a[i+1];
9             if(j > k)
10              {iTmp = a[i];
11                a[i] = a[i+1];
12                a[i+1] = iTmp;
13                sorted = false;
14              }
15            }
16          }
17        }
```

- A) What function do lines 10, 11, and 12 perform?
- B) What is the minimum number of times that the while loop beginning in line 4 will run. Consider all cases including the case where a is already sorted. Justify and explain your answer.
- C) Does this sort method sort data in ascending order (a to z) or descending order (z to a)?
- D) What is the purpose of the boolean variable sorted?
- E) Show how to activate the sort method to sort the array defined in a main program by:
`int [] xData = {5, 6, 4, 8, 2, 9, 1, 7, 3, 2};`

8. The following method does a *Select sort* on a parameter named *arr*. Line numbers have been added to the left for reference purposes. Answer the questions below with regard to this method.

```
1 private void Sort(int [] arr)
2 {int i, j, tmp, minIndx;
3   for(i=0;i<arr.Length-1;i++)
4     {for(j=i;j<arr.Length;j++)
5       {FindMin(arr, i, out minIndx);
6         tmp = arr[i];
7         arr[i] = arr[minIndx];
8         arr[minIndx] = tmp;
9       }
10    }
11 }
12 private void FindMin(int [] data, int start, out int minIndx)
13 {int i;
14   minIndx = start;
15   for(i=start;i<data.Length;i++)
16     {if(data[i] < data[minIndx])
17       minIndx = i;
18     }
19 }
```

- A) What do lines 6, 7, and 8 do?
- B) What is the minimum number of times that the for loop beginning in line 3 will run if the array being sorted has 15 elements and is already in sorted order.
- C) If *arr* has just 3 elements what is the maximum number of times the for loop beginning in line 3 will run. Justify and explain your answer.
- D) What is the purpose of the *start* variable in the *FindMin* method?
- E) If a main program has an array of ints created by
`int [] mArr = new int[1024];`
and the array has been filled with data, show how to sort this array using the *Sort* method above.

9. A GUI application has two text boxes txt1 and txt2 and a button btn1. Write a program which will input a number from the first text box and add it to the total in the second text box each time the button is pushed. The second text box will have the sum of all of the numbers entered in the first text box.

```
private void btn1_Click(object sender, EventArgs e)
{
```

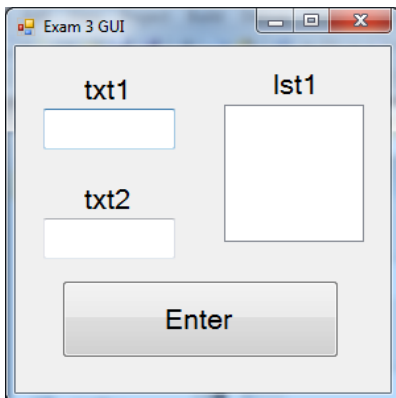
10. In the GUI below there are two text boxes named txt1 and txt2, a list box named lst1, and a button named btn1. Write the button click event to do the following:

A) Read the text in txt1. If the txt length is 0 return. If the text length is greater than 0 create a character variable called *c* which is equal to the first character in the text.

B) If *c* is a digit add it to the sum of the integers in lst1 and add the new sum to the list box.

C) If *c* is a character other than a digit clear txt2 and write the character to txt2.

The last entry in the list box will always be the sum of all of the first digits in txt1.



```
private void btn1_Click(object sender, EventArgs e)
{
```