

1.1P: Preparing for OOP – Answer Sheet

Introduction

This answer sheet serves two purposes:

- A. It serves as a revision for you of your previous learnings; and
- B. It establishes a baseline understanding of your knowledge in key Computer Science topics.

As such, this answer sheet is divided into the following areas of knowledge:

- A. Your experience with UNIX/DOS console commands;
- B. Your ability to differentiate between data types (e.g., text) and information categories (e.g., title);
- C. Your experience with parsing and evaluating expressions according to rules of precedence;
- D. Your understanding of computer science concepts and various programming language constructs;
- E. Finally we want you to develop a simple function called *Average*. You will develop a fully functional program in three steps:
 - 1) Implement the *Average* function,
 - 2) Define a main function calling *Average*, and
 - 3) Define tests and output the result of calling *Average* on a given array of numbers.

Section A: Console commands

Explain the following terminal instructions:

- a) ***cd***:

This command is used to change the directory.

- b) ***pwd***:

This command prints the full name of the current directory.

c) ***mkdir***.

This command is used to make a new directory.

d) ***cat***:

The cat command reads each file parameter in sequence and writes it to standard output.

e) ***ls***:

This command is used to list names and features of files and directories.

Section B: Data types and Information Classes

1. Consider the following kinds of information, and suggest the most appropriate data type to store or represent each class of information:

Information Category	Suggested Data Type
A person's family name	String
A person's age in years	Integer
A person's weight in kilograms	Float
A telephone number	Integer
A temperature on the Kelvin scale	Float

The average age of a group of children	Integer
Whether a student has passed this task	Boolean

2. Aside from the examples already provided in question 1, consider the following list and find an example of information that could be stored as:

Data type	Suggested Information Category
String	A schools name
Integer	Date of birth
Float	Test score
Boolean	Whether a light is on or off

Section C: Parsing and Evaluating Expressions

Fill out the **last** two columns of the following table. Parse and evaluate each expression. Enter the resulting value in column 3 and specify, in column 4, the data type of the expression in a compiler “friendly” form (i.e., in a language format accepted, for example, in C, C#, or Pascal):

Expression	Given	Value	Data Type
6		6	Integer
True		True	Boolean

A	a = 2.5	2.5	Float
1 + 2 * 3		7	Integer
a and False	a = True	False	Boolean
a or False	a = True	True	Boolean
a + b	a = 1 b = 2	3	Integer
2 * a	a = 3	6	Integer
a * 2 + b	a = 2.5 b = 3	8	Integer
a + 2 * b	a = 2.5 b = 3	8.5	Float
(a + b) * c	a = 1 b = 2 c = 3	9	Integer
"Fred" + " Flintstone"		Fred Flintstone	String
a + " Rubble"	a = "Barney"	Rubble Barney	String

Section D: Computer Science and Programming Language Concepts:

1. Using an example, explain the difference between **declaring** and **initializing** a variable.

Declaring is to specify the type of data for the variable whereas initializing is to set the value of the variable to its initials.

Insert your example here:

```
2
3   int x; //this is declaring a variable
4   int x = 10; // this is initializing a variable
5
```

2. Explain the concept *parameter*. Write some code that demonstrates a simple use of a parameter. You should show a procedure or function that uses a parameter, and how you would call that procedure or function.

A parameter is a variable or value that is passed into a function or method when it is called.

Insert your example here:

```
1   using System;
2
3   public class Program
4   {
5       public static void Parameter(string name, int age) // name and age are two parameters
6       {
7           Console.WriteLine("hello " + name + " ,you are " + age);
8       }
9       public static void Main()
10      {
11          Console.WriteLine("enter name");
12          string name = Console.ReadLine();
13          Console.WriteLine("enter age");
14          int age = Convert.ToInt32(Console.ReadLine());
15
16          Parameter(name, age); // in this case name and age are arguments
17      }
18  }
19
20
```

3. Using code examples, describe the term *scope* as it is being used in programming (not in business or project management). In your explanation, focus on procedural programming. Ensure that you cover as many kinds of scopes and their respective differences as possible. Your answer must detail at least two kinds of scopes.

Scope is a region where the variable is defined and can be accessed.

Global scope : Variable declared in the global scope are accessed in the entire program.

Local scope: Variable declared in Local scope are accessed only in that function.

Insert your examples here:

```
1 using System;
2
3 public class Program
4 {
5     //global scope
6     static int globalVariable = 10;
7
8     public static void Main()
9     {
10         Console.WriteLine($"{ globalVariable}"); // accessing global variable
11
12         //declaring local variable within the main function
13         int mainLocalVariable = 20;
14         Console.WriteLine($"{mainLocalVariable}"); //accessing Local variable within the main function
15         //calling a function with its own local variable
16         NewFunction();
17     }
18     public static void NewFunction()
19     {
20         int localVariable = 30;
21         Console.WriteLine($"{localVariable}"); //accessing Local variable within NewFunction
22         //accessing global variable within the function
23
24         Console.WriteLine($"{globalVariable}");
25     }
26 }
27
28
```

Terminal - ctest101

```
10
20
30
10
```

Section E: Programming Practice:

1. Using procedural style programming, in any language you like, write a function called *Average*, which accepts an array of integers and returns the average of those integers.

Do not use any libraries for calculating the average: we want to see your understanding of algorithms. You must demonstrate appropriate use of parameters, returning and assigning values, and the use of control statements like a loop. Note — just write the function `Average` at this point. In the next question we will ask you to *invoke the Average function*. You are not required to develop a complete program or even specify code that outputs anything at this stage. `Average` is a pure function. Input/output and any business logic processing is the responsibility of the (main line) calling the function `Average`.

Insert your code here:

```
10 public static int Average(int[] arr)
11 {
12     int sum = 0;
13     foreach (int i in arr)
14     {
15         sum += i;
16     }
17     int average = sum / arr.Length;
18     return average;
19 }
20
21 }
```

2. Using the same language, write a main function you would to set up data (i.e., declare an array and initialize it with proper values), call the `Average` function, and print out the result. You are not required to provide input processing logic; you can have an inline instantiate of the collection of data values that the `Average` function needs to calculate the average of. Please use a reasonable data set, that is, the array must contain at least five elements of numerical type.

Insert your code here:


```

2  class Program
3  {
4      public static void Main(string[] args)
5      {
6          int[] arr = { 10, 20, 30, 40, 50 };
7          int ans = Average(arr);
8          Console.WriteLine("the average is " + ans);
9      }
10
11     public static int Average(int[] arr)
12     {
13         int sum = 0;
14         foreach (int i in arr)
15         {
16             sum += i;
17         }
18         int average = sum / arr.Length;
19         return average;
20     }
21 }
22

```

- Again using the same language, extend the main function with some output statements. Print the message "Double digits" if the average is above or equal to 10. Otherwise, print the message "Single digits". And then, if the average is negative (e.g., the average of a week's temperature readings at the Australian base in the Antarctic), add an additional line of output highlighting that the "Average value is in the negative". Provide screenshot(s) of your program running, that is, the code and the run time output.

Insert your code here:

```
1  using System;
2  class Program
3  {
4      public static void Main(string[] args)
5      {
6          int[] arr = { 10, 20, 30, -40, -50 };
7          float ans = Average(arr);
8          Console.WriteLine("the average is " + ans);
9
10         // check double digits or not
11         if (ans < 0)
12         {
13             Console.WriteLine("Average value is in the negative");
14         }
15         else if (ans < 10)
16         {
17             Console.WriteLine("Single Digit");
18         }
19         else
20         {
21             Console.WriteLine("Double Digit");
22         }
23     }
24
25     public static float Average(int[] arr)
26     {
27         int sum = 0;
28         foreach (int i in arr)
29         {
30             sum += i;
31         }
32     }
33 }
```

Terminal - Average

```
the average is -6
Average value is in the negative
```

Build Output

```
Build succeeded.
0 Warning(s)
0 Error(s)

Time Elapsed 00:00:00.14

===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====

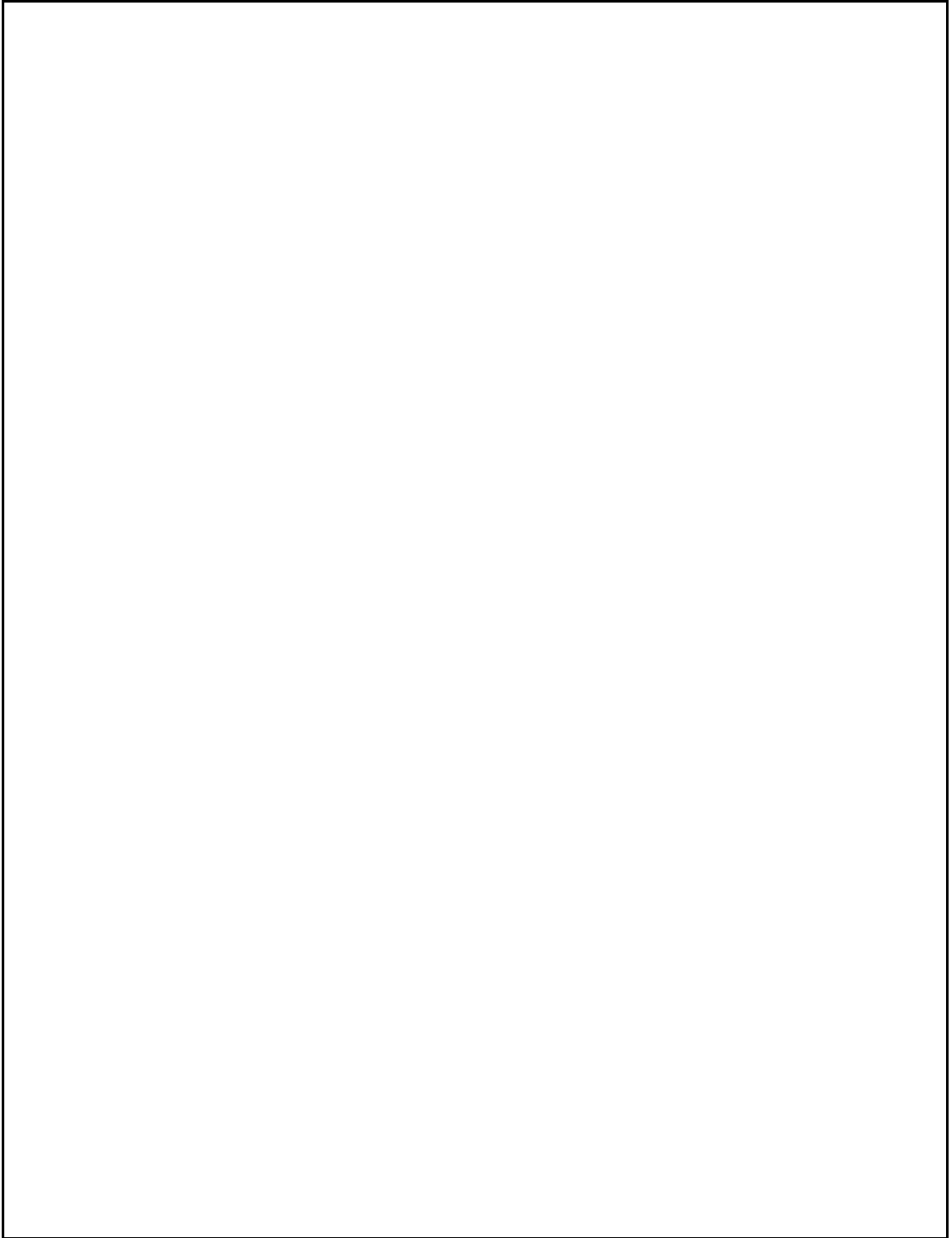
Build successful.
```

Insert your whole program here:

```

1  using System;
2  class Program
3  {
4      public static void Main(string[] args)
5      {
6          int[] arr = { 10, 20 ,30 ,-40,-50 };
7          float ans = Average(arr);
8          Console.WriteLine("the average is "+ ans);|
9
10         // check double digits or not
11         if (ans < 0)
12         {
13             Console.WriteLine("Average value is in the negative");
14         }
15         else if (ans < 10)
16         {
17             Console.WriteLine("Single Digit");
18         }
19         else
20         {
21             Console.WriteLine("Double Digit");
22         }
23     }
24
25     public static float Average(int[] arr)
26     {
27         int sum = 0;
28         foreach (int i in arr)
29         {
30             sum += i;
31         }
32         float average = sum / arr.Length;
33         return average;
34     }
35 }
36

```



End of Task

All students have access to the Adobe Acrobat tools. Please print your solution to PDF and submit via Canvas.