This assignment introduces loops (Chapter 5), methods (Chapter 6), and arrays (Chapter 7). Read these three chapters.

Methods are programming structures also called procedures, functions, or subroutines in other languages, although these words do not always refer to exactly the same structures. Parameters/attributes (essentially the data) and variables are the other major components of a class.

For security purposes it is recommended that all attributes are declared private so that only the methods within a class can change their values, which is best done by set() methods after the values are tested for being valid. The best way to do the validation is with exceptions and will be considered in Assignment 4.  Arrays are actually a form of parameters that can be considered as tables with rows and columns in which many values are named by a single variable name.

The use of programs often benefits by allowing the user to supply input, then echo the input along with the output, and ask if the user wants to run the program **again** with new input.  This allows the programs to be tested with many possible input combinations, both valid and not, without exiting.

Repeat loops can be implemented by embedding the program in one overall loop that cycles until the user indicates the program should exit.  At the conclusion of each loop the program needs to prompt for exit or not.  One way to implement repeat loops as part of the user-computer dialog is described in the document: [CS501Spring2017UserDialogRulesB.txt](https://sit.instructure.com/courses/26944/files/3969864/download?verifier=L8bM1u9YclNOkTIayAdccRuns80J93apx1VKpnr7&wrap=1).  Start using repeat loops for all exercises from now on except those for which they would be a nuisance.  Assignment instructions should indicate when an exercise should not call for user inputs and you are welcome to always ask.

|  |  |  |
| --- | --- | --- |
| Assignment 3 | 11th Edition | 10th Edition |
| CS | 5.16/5.20/6.10 as a single program and 7.9/ 7.10/7.20 as a single program (see notes below) | 5.16/5.20/6.10 as a single program and 7.9/ 7.10/7.20 as a single program (see notes below) |
| MIS | 5.22, 6.11, 7.9/7.10/7.20 as a single program (see notes below) | 5.22, 6.11, 7.9/7.10/7.20 as a single program (see notes below) |

 For prime number exercises 5.16/5.20/6.10 10th show

* only the first 50, then
* populate an array containing the prime numbers less than 1000, and
* in a repeat loop ask the user to enter one number at a time to test if it is prime  by searching the array.  If not show its factors.  Searching arrays is described in Chapter 7.

For 7.9/7.10/7.20 and 6.9/6.10/6.20 as a single program the idea is to have the user enter the 10 numbers only once.  First in 7.9/7.10 or 6.9/6.10 get the minimum and its index, report it, then the sort the list and show it.

Remember:

1. All programs need a brief introduction showing what the program will do (intro statement).
2. Then show prompts to the user for the inputs required for the program.  Need to say what each input is (*e.g.,* letter, word, number, phrase), its units,  and how it should be formatted (*e.g.,* 3 numbers separated by spaces).
3. If appropriate, a repeat loop so that when the first set of input is done the programs ask the user if it should run again for more, presumably different input. There should be an option to exit or to repeat the program with new input.