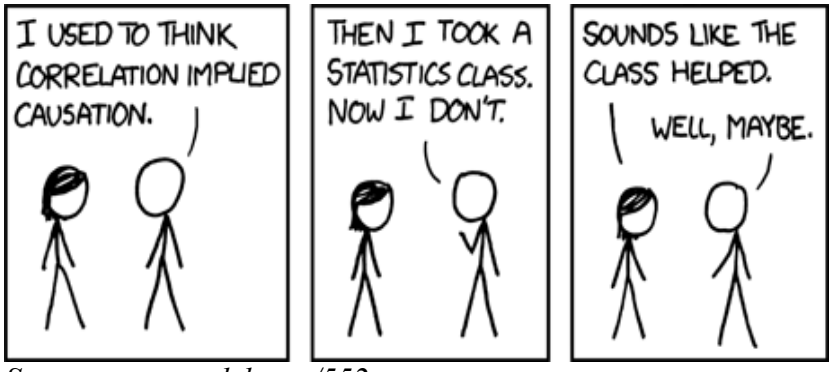
**CS 202**
**Assignment #10**

Purpose: Learn to use C++ templates and perform exception handling
Due: Tuesday (4/02)
Points: 125

**Assignment:**

A data set[1] (or dataset) is a collection of data. Each value is referred to as a datum. When performing statistical operations, the data sets may vary in type (integers, shorts, floats, doubles, strings, etc.).



*Source: www.xckd.com/552*

Develop a class, using templates, to provide functionality for a series of basic and advanced statistical operations. The UML class specifications for a polymorphic statics package are provided below. A main will be provided that uses the ***statisticsPkg*** class.

- Statistic Package Class
  The statistics package template class should implement the follow template functions.

| statisticsPkg |
|---|
| -setLength: int |
| -*mySet: myType |
| +CNT_MIN=5, CNT_MAX=9999: static const int |
| +RND_LIMIT=999: static const int |
| +statisticsPkg() |
| +statisticsPkg(int, myType[]) |
| +~statisticsPkg() |
| +getDatum(int) const: myType |
| +setDatum(int, myType): void |
| +readCount(): int |
| +generateNewSet(int): void |
| +printSet() const: void |
| +minimum() const: myType |
| +maximum() const: myType |
| +median() const: myType |
| +sum() const: myType |
| +average() const: myType |
| +standardDeviation() const: myType |
| +shellSort(): void |

1 For more information, refer to: http://en.wikipedia.org/wiki/Data_set

| +coVariance(const statisticsPkg&) const: myType |
|---|
| +pearsonCoefficient(const statisticsPkg&) const: myType |
| +linearCorrelationCoefficient(const statisticsPkg&) const: myType |
| -freeMemory(): void |

*Note*, points will be deducted for insufficient commenting, poor style or inefficient coding.

**Function Descriptions**

- The *statisticsPkg()* constructor should set the *setLength* to 0 and *mySet* point to NULL. The *statisticsPkg(int, myType[])* constructor should accept an integer for the size and create a single dimension array of *myType* to hold items in the passed data set. The constructor should ensure the array is valid (i.e., not NULL) and that the size is between CNT_MIN and CNT_MAX. If there is an error, an appropriate message should be displayed and the class variables set to the default values (0 and NULL). If there are not errors, the constructor should dynamically create the new class array and copy the items from the passed array to the newly created class array.

- The *~statisticsPkg()* destructor calls the *freeMemory()* function which should delete the *myType* array, set the pointer to NULL, and set the size to 0.

- The *shellSort()* function sorts the data set with the shell sort (see below). Must use provided shell sort algorithm. *Note*, use of other sort algorithms will result in zero score.

- The *generateNewSet(int)* function should generate a new set which may require removal of the current set (via a call to the *freeMemory()* function). The passed length should be verified and, if valid, the *setLength* should set *rand()* used to generate the data. The generated data item values should be limited in range to 0 and RND_LIMIT-1 (via a mod operation) as follows: **`static_cast <myType> (rand()%RND_LIMIT)`**

- The *maximum()* function should find and return the maximum value in the set. The *minimum()* function should find and return the minimum value in the set. Theses functions do not require the data to be sorted. The *median()* function should find and return the median value in the set. The function should assume the data is sorted. *Note,* for an odd number of items, the median value is defined as the middle value. For an even number of values, it is the average of the two middle values.

- The *getDatum(int)* function should return the data item of type *myType* at the location of the passed index (integer). The *setDatum(int, myType)* function set the passed data item of type *myType* at the location of the passed index. Both functions must ensure the index is legal (within range and the specific data set) and provide an error if not.

- The *printSet()* function print data items of type *myType*, including the index and value at *myType*. The values should be displayed 3 per line in the format as shown in the example. *Note*, using **`showpoint`** on non-real data types is ignored.

- The *readCount()* function should read the count of data items (to generate). The function should ensure that the value is between CNT_MIN and CNT_MAX, inclusive. If an errors occurs (out of range or invalid input) an appropriate message should be displayed and the user re-prompted. The function should limit the incorrect entries to four tries. Specifically, three errors is acceptable, but a fourth error should end the function and return 0 for the count. The function must trap invalid input (including character input).

- The *standardDeviation()* function should compute the standard deviation[2] of the current class data set (if one exists). The formula for standard deviation is:

$$standardDeviation = \sqrt{\frac{\sum_{i=0}^{length-1} \left(x[i] - \bar{x}\right)^2}{length}}$$

Where $\bar{x}$ is the average of the elements in the ***x*** array values.

- The *coVariance(statisticsPkg &)* function should compute the covariance[3] between two data sets. The formula for covariance is as follows:

$$coVariance = \frac{\sum_{i=0}^{length-1} \left(x[i] - \bar{x}\right)\left(y[i] - \bar{y}\right)}{length - 1}$$

Where $\bar{x}$ is the average of the ***x*** array values, $\bar{y}$ is the average of the ***y*** array values, $\sum$ is the summation of the values, and ***length*** is the count of data items.

- The *perasonsCoefficient(statisticsPkg &)* function should return the Pearson's product-moment coefficient[4] between passed data set and the class data set (which must be of the same size). The formula for the correlation coefficient is as follows.

$$perasonsCoefficient = \frac{\sum_{i=0}^{length-1} \left(x[i]\,y[i] - length\ \bar{x}\ \bar{y}\right)}{\left(length-1\right) s_x\ s_y}$$

Where $\bar{x}$ is the average of the ***x*** array values, $\bar{y}$ is the average of the ***y*** array values, $\sum$ is the summation of the values, $s_x$ and $s_y$ are the standard deviation of the ***x*** and ***y*** arrays, and ***length*** is the count of data items.

- The *linaearCorrelationCoefficient(statisticsPkg &)* function is the linear correlation coefficient between two data sets. The linear correlation coefficient is also referred to as the Peason's correlation coefficient for a population. For similar data sets, the Pearson's product-moment correlation coefficient and the linear correlation coefficient may be the same value. For formula for the linear correlation coefficient is as follows:

$$linearCorrelationCoefficient = \frac{coVar(x, y)}{s_x\ s_y}$$

Where ***coVar(x,y)*** is the covariance between the ***x*** and ***y*** arrays and $s_x$ and $s_y$ are the standard deviation of the ***x*** and ***y*** arrays.

Refer to the example executions for output formatting. Make sure your program includes the appropriate documentation. See Program Evaluation Criteria for CS 202 for additional information.

---

2 For more information, refer to: http://en.wikipedia.org/wiki/Standard_deviation
3 For more information, refer to: http://en.wikipedia.org/wiki/Covariance
4 For more information, refer to: http://en.wikipedia.org/wiki/Correlation_and_dependence

**Shell Sort**
To sort the numbers, use the following Shell Sort[5] algorithm:

```
h = 1
while ( (h*3+1) < length ) {
    h = 3 * h + 1
}

while ( h>0 ) {
    for (i = h-1; i < length; i++) {
        tmp = lst[i]
        for ( j=i; (j>=h) && (lst[j-h]>tmp); j = j-h)
            lst[j] = lst[j-h]
        lst[j] = tmp
    }
    h = h / 3
}
```

There are many variations of the Shell Sort algorithm. You must use the above Shell Sort algorithm (i.e., do not use a different sort). The algorithm assumes array index's start at 0. ***Submissions not based on this algorithm will not be scored***.

**Make File:**
You will need to develop a make file. You should be able to type:

```
make
```

Which should create the executable. The makefile will be similar to the previous assignment makefiles.

**Submission:**

- Submit a compressed zip file of the program source files, header files, and makefile via the on-line submission by 23:50.

All necessary files must be included in the ZIP file. The grader will download, uncompress, and type **make** (so you must have a valid, working *makefile*).

**Example Execution:**
Below is an example program execution for the main.

Example #1:

```
ed-vm% ./main
======================================================================
Assignment #10 -  Statistics Package -> Testing

**********************************************************************
Data set - Doubles

Data Set A for double type (unsorted):
 Set[0] =  2.50000      Set[1] =  4.50000      Set[2] =  6.50000
 Set[3] =  9.50000      Set[4] =  10.5000      Set[5] =  3.50000
 Set[6] =  8.50000      Set[7] =  5.50000      Set[8] =  7.50000
 Set[9] =  1.50000

Data Set B for double type (unsorted):
```

5 For more information, refer to: http://en.wikipedia.org/wiki/Shellsort

```
 Set[0] =  12.5000      Set[1] =  18.5000      Set[2] =  10.5000
 Set[3] =  13.5000      Set[4] =  19.5000      Set[5] =  15.5000
 Set[6] =  16.5000      Set[7] =  17.5000      Set[8] =  11.5000
 Set[9] =  14.5000

------------------------------------------------------------------------
Data Set A for double type (sorted):
 Set[0] =  1.50000      Set[1] =  2.50000      Set[2] =  3.50000
 Set[3] =  4.50000      Set[4] =  5.50000      Set[5] =  6.50000
 Set[6] =  7.50000      Set[7] =  8.50000      Set[8] =  9.50000
 Set[9] =  10.5000

Statistical Results
   Data Set A Minimum: 1.50000
   Data Set A Maximum: 10.5000
   Data Set A Median: 6.00000
   Data Set A Sum: 60.0000
   Data Set A Average: 6.00000
   Data Set A Standard Deviation: 2.87228

------------------------------------------------------------------------
Data Set B for double type (sorted):
 Set[0] =  10.5000      Set[1] =  11.5000      Set[2] =  12.5000
 Set[3] =  13.5000      Set[4] =  14.5000      Set[5] =  15.5000
 Set[6] =  16.5000      Set[7] =  17.5000      Set[8] =  18.5000
 Set[9] =  19.5000

Statistical Results
   Data Set B Minimum: 10.5000
   Data Set B Maximum: 19.5000
   Data Set B Median: 15.0000
   Data Set B Sum: 150.000
   Data Set B Average: 15.0000
   Data Set B Standard Deviation: 2.87228

------------------------------------------------------------------------
Summary Results:
   Covariance: 9.16667
   Pearson's Coefficient: 1.11111
   Linear Correlation Coefficient: 1.11111


************************************************************************
Data set - Integers

Data Set A for integer type (unsorted):
 Set[0] =     1234      Set[1] =      664      Set[2] =      153
 Set[3] =      268      Set[4] =      500      Set[5] =      997
 Set[6] =      991      Set[7] =      903      Set[8] =      762
 Set[9] =      253      Set[10] =      590

Data Set B for integer type (unsorted):
 Set[0] =     4321      Set[1] =      842      Set[2] =      682
 Set[3] =      707      Set[4] =      409      Set[5] =       87
 Set[6] =      351      Set[7] =      565      Set[8] =      496
 Set[9] =      251      Set[10] =      485

------------------------------------------------------------------------
Data Set A for integer type (sorted):
 Set[0] =      153      Set[1] =      253      Set[2] =      268
 Set[3] =      500      Set[4] =      590      Set[5] =      664
 Set[6] =      762      Set[7] =      903      Set[8] =      991
 Set[9] =      997      Set[10] =     1234

Statistical Results
   Data Set A Minimum: 153
   Data Set A Maximum: 1234
   Data Set A Median:  664
   Data Set A Sum: 7315
   Data Set A Average: 665
   Data Set A Standard Deviation: 334

------------------------------------------------------------------------
Data Set B for integer type (sorted):
 Set[0] =       87      Set[1] =      251      Set[2] =      351
```

```
 Set[3] =      409      Set[4] =      485      Set[5] =      496
 Set[6] =      565      Set[7] =      682      Set[8] =      707
 Set[9] =      842      Set[10] =     4321

Statistical Results
   Data Set B Minimum: 87
   Data Set B Maximum: 4321
   Data Set B Median:  496
   Data Set B Sum: 9196
   Data Set B Average: 836
   Data Set B Standard Deviation: 1120

---------------------------------------------------------------------
Summary Results:
   Covariance: 279413
   Pearson's Coefficient: 0
   Linear Correlation Coefficient: 0


*********************************************************************
Data set - Floats

Data Set A for float type (unsorted):
 Set[0] =  3.14000      Set[1] =  114.000      Set[2] =  584.000
 Set[3] =  413.000      Set[4] =  863.000      Set[5] =  22.0000
 Set[6] =  388.000      Set[7] =  307.000      Set[8] =  545.000
 Set[9] =  585.000      Set[10] =  972.000      Set[11] =  417.000
 Set[12] =  572.000      Set[13] =  192.000      Set[14] =  415.000
 Set[15] =  565.000      Set[16] =  814.000      Set[17] =  178.000
 Set[18] =  537.000      Set[19] =  405.000      Set[20] =  765.000
 Set[21] =  380.000      Set[22] =  806.000      Set[23] =  193.000
 Set[24] =  509.000

Data Set B for float type (unsorted):
 Set[0] =  6.28000      Set[1] =  263.000      Set[2] =  75.0000
 Set[3] =  110.000      Set[4] =  514.000      Set[5] =  280.000
 Set[6] =  674.000      Set[7] =  629.000      Set[8] =  864.000
 Set[9] =  806.000      Set[10] =  212.000      Set[11] =  886.000
 Set[12] =  913.000      Set[13] =  519.000      Set[14] =  432.000
 Set[15] =  500.000      Set[16] =  492.000      Set[17] =  569.000
 Set[18] =  791.000      Set[19] =  403.000      Set[20] =  984.000
 Set[21] =  76.0000      Set[22] =  218.000      Set[23] =  882.000
 Set[24] =  333.000

---------------------------------------------------------------------
Data Set A for float type (sorted):
 Set[0] =  3.14000      Set[1] =  22.0000      Set[2] =  114.000
 Set[3] =  178.000      Set[4] =  192.000      Set[5] =  193.000
 Set[6] =  307.000      Set[7] =  380.000      Set[8] =  388.000
 Set[9] =  405.000      Set[10] =  413.000      Set[11] =  415.000
 Set[12] =  417.000      Set[13] =  509.000      Set[14] =  537.000
 Set[15] =  545.000      Set[16] =  565.000      Set[17] =  572.000
 Set[18] =  584.000      Set[19] =  585.000      Set[20] =  765.000
 Set[21] =  806.000      Set[22] =  814.000      Set[23] =  863.000
 Set[24] =  972.000

Statistical Results
   Data Set A Minimum: 3.14000
   Data Set A Maximum: 972.000
   Data Set A Median: 417.000
   Data Set A Sum: 11544.1
   Data Set A Average: 461.766
   Data Set A Standard Deviation: 254.533

---------------------------------------------------------------------
Data Set B for float type (sorted):
 Set[0] =  6.28000      Set[1] =  75.0000      Set[2] =  76.0000
 Set[3] =  110.000      Set[4] =  212.000      Set[5] =  218.000
 Set[6] =  263.000      Set[7] =  280.000      Set[8] =  333.000
 Set[9] =  403.000      Set[10] =  432.000      Set[11] =  492.000
 Set[12] =  500.000      Set[13] =  514.000      Set[14] =  519.000
 Set[15] =  569.000      Set[16] =  629.000      Set[17] =  674.000
 Set[18] =  791.000      Set[19] =  806.000      Set[20] =  864.000
 Set[21] =  882.000      Set[22] =  886.000      Set[23] =  913.000
 Set[24] =  984.000
```

```
Statistical Results
   Data Set B Minimum: 6.28000
   Data Set B Maximum: 984.000
   Data Set B Median: 500.000
   Data Set B Sum: 12431.3
   Data Set B Average: 497.251
   Data Set B Standard Deviation: 291.746

----------------------------------------------------------------------
Summary Results:
   Covariance: 75218.1
   Pearson's Coefficient: 1.01292
   Linear Correlation Coefficient: 1.01292


**********************************************************************
Data set - chars

Data for char type (unsorted):
 Set[0] =        y      Set[1] =        u      Set[2] =        a
 Set[3] =        e      Set[4] =        i      Set[5] =        o


Data for char type (sorted):
 Set[0] =        a      Set[1] =        e      Set[2] =        i
 Set[3] =        o      Set[4] =        u      Set[5] =        y


**********************************************************************
Data set - chars

Data for string type (unsorted):
 Set[0] =    Black      Set[1] =    Brown      Set[2] =      Red
 Set[3] =   Orange      Set[4] =   Yellow      Set[5] =    Green
 Set[6] =     Blue      Set[7] =   Violet      Set[8] =     Grey
 Set[9] =    White

Data for string type (sorted):
 Set[0] =    Black      Set[1] =     Blue      Set[2] =    Brown
 Set[3] =    Green      Set[4] =     Grey      Set[5] =   Orange
 Set[6] =      Red      Set[7] =   Violet      Set[8] =    White
 Set[9] =   Yellow

**********************************************************************
Data set - Long Long

Data for long long type (unsorted):
 Set[0] =      342      Set[1] =      648      Set[2] =      713
 Set[3] =      150      Set[4] =      560      Set[5] =      941
 Set[6] =      762

Data Set Minimum: 300
Data Set Maximum: 1882
Data Set Median: 1296
Data Set Sum: 8232
Data Set Average: 1176
Data Set Standard Deviation: 494

----------------------------------------------------------------------

Data for long long type (sorted):
 Set[0] =      300      Set[1] =      684      Set[2] =     1120
 Set[3] =     1296      Set[4] =     1426      Set[5] =     1524
 Set[6] =     1882

**********************************************************************
Error Testing - Integer Type

Error (constructor), invalid set length.
Error (constructor), invalid set length.
Error (constructor), invalid set length.
Error (constructor), invalid input array.
Error (setDatum), index out or range.
Error (setDatum), index out or range.
```

```
Error (getDatum), index out or range.
Error (getDatum), index out or range.
Error (generate), invalid set length.
Error (generate), invalid set length.
Error (generate), invalid set length.
Error (coVariance), invalid data set sizes.
Error (coVariance), invalid data set sizes.
Error (perasonsCoefficient), invalid data set sizes.
Error (perasonsCoefficient), invalid data set sizes.
Error (linearCorrelationCoefficient), invalid data set sizes.
Error (linearCorrelationCoefficient), invalid data set sizes.

---------------------------------------------------------------------

Data for int type (sorted):
 Set[0] =       17      Set[1] =      339      Set[2] =      591
 Set[3] =      736      Set[4] =      824

*********************************************************************
Data set - Shorts

Enter Count (5-9999): -3

Error (read), count Value -3 not between 5 and 9999.
Enter Count (5-9999): three

Error, count out of range.
Enter Count (5-9999): 7


Data Set A for short type (unsorted):
 Set[0] =       26      Set[1] =       68      Set[2] =       60
 Set[3] =        7      Set[4] =       89      Set[5] =       48
 Set[6] =       98

Data Set B for short type (unsorted):
 Set[0] =       42      Set[1] =       63      Set[2] =       20
 Set[3] =       63      Set[4] =       20      Set[5] =       71
 Set[6] =        3

---------------------------------------------------------------------
Data Set A for short type (sorted):
 Set[0] =        7      Set[1] =       26      Set[2] =       48
 Set[3] =       60      Set[4] =       68      Set[5] =       89
 Set[6] =       98

Data Set Minimum: 7
Data Set Maximum: 98
Data Set Median:  60
Data Set Sum:     396
Data Set Average: 56
Data Set Standard Deviation: 30

---------------------------------------------------------------------
Data Set B for short type (sorted):
 Set[0] =        3      Set[1] =       20      Set[2] =       20
 Set[3] =       42      Set[4] =       63      Set[5] =       63
 Set[6] =       71

Data Set Minimum: 3
Data Set Maximum: 71
Data Set Median:  42
Data Set Sum:     282
Data Set Average: 40
Data Set Standard Deviation: 24

---------------------------------------------------------------------
   Covariance: 819
   Pearson's Coefficient: 1
   Linear Correlation Coefficient: 1

ed-vm%
```

Example #2:

```
ed-vm% ./main
========================================================================
Assignment #10 -  Statistics Package -> Testing

************************************************************************
Data set - Doubles

Data Set A for double type (unsorted):
 Set[0] =  2.50000      Set[1] =  4.50000      Set[2] =  6.50000
 Set[3] =  9.50000      Set[4] =  10.5000      Set[5] =  3.50000
 Set[6] =  8.50000      Set[7] =  5.50000      Set[8] =  7.50000
 Set[9] =  1.50000

Data Set B for double type (unsorted):
 Set[0] =  12.5000      Set[1] =  18.5000      Set[2] =  10.5000
 Set[3] =  13.5000      Set[4] =  19.5000      Set[5] =  15.5000
 Set[6] =  16.5000      Set[7] =  17.5000      Set[8] =  11.5000
 Set[9] =  14.5000

------------------------------------------------------------------------
Data Set A for double type (sorted):
 Set[0] =  1.50000      Set[1] =  2.50000      Set[2] =  3.50000
 Set[3] =  4.50000      Set[4] =  5.50000      Set[5] =  6.50000
 Set[6] =  7.50000      Set[7] =  8.50000      Set[8] =  9.50000
 Set[9] =  10.5000

Statistical Results
   Data Set A Minimum: 1.50000
   Data Set A Maximum: 10.5000
   Data Set A Median: 6.00000
   Data Set A Sum: 60.0000
   Data Set A Average: 6.00000
   Data Set A Standard Deviation: 2.87228

------------------------------------------------------------------------
Data Set B for double type (sorted):
 Set[0] =  10.5000      Set[1] =  11.5000      Set[2] =  12.5000
 Set[3] =  13.5000      Set[4] =  14.5000      Set[5] =  15.5000
 Set[6] =  16.5000      Set[7] =  17.5000      Set[8] =  18.5000
 Set[9] =  19.5000

Statistical Results
   Data Set B Minimum: 10.5000
   Data Set B Maximum: 19.5000
   Data Set B Median: 15.0000
   Data Set B Sum: 150.000
   Data Set B Average: 15.0000
   Data Set B Standard Deviation: 2.87228

------------------------------------------------------------------------
Summary Results:
   Covariance: 9.16667
   Pearson's Coefficient: 1.11111
   Linear Correlation Coefficient: 1.11111


************************************************************************
Data set - Integers

Data Set A for integer type (unsorted):
 Set[0] =     1234      Set[1] =      664      Set[2] =      153
 Set[3] =      268      Set[4] =      500      Set[5] =      997
 Set[6] =      991      Set[7] =      903      Set[8] =      762
 Set[9] =      253      Set[10] =      590

Data Set B for integer type (unsorted):
 Set[0] =     4321      Set[1] =      842      Set[2] =      682
 Set[3] =      707      Set[4] =      409      Set[5] =       87
 Set[6] =      351      Set[7] =      565      Set[8] =      496
 Set[9] =      251      Set[10] =      485

------------------------------------------------------------------------
Data Set A for integer type (sorted):
```

```
 Set[0] =      153      Set[1] =      253      Set[2] =      268
 Set[3] =      500      Set[4] =      590      Set[5] =      664
 Set[6] =      762      Set[7] =      903      Set[8] =      991
 Set[9] =      997      Set[10] =     1234

Statistical Results
   Data Set A Minimum: 153
   Data Set A Maximum: 1234
   Data Set A Median:  664
   Data Set A Sum: 7315
   Data Set A Average: 665
   Data Set A Standard Deviation: 334

------------------------------------------------------------------------
Data Set B for integer type (sorted):
 Set[0] =       87      Set[1] =      251      Set[2] =      351
 Set[3] =      409      Set[4] =      485      Set[5] =      496
 Set[6] =      565      Set[7] =      682      Set[8] =      707
 Set[9] =      842      Set[10] =     4321

Statistical Results
   Data Set B Minimum: 87
   Data Set B Maximum: 4321
   Data Set B Median:  496
   Data Set B Sum: 9196
   Data Set B Average: 836
   Data Set B Standard Deviation: 1120

------------------------------------------------------------------------
Summary Results:
   Covariance: 279413
   Pearson's Coefficient: 0
   Linear Correlation Coefficient: 0


************************************************************************
Data set - Floats

Data Set A for float type (unsorted):
 Set[0] =  3.14000      Set[1] =  114.000      Set[2] =  584.000
 Set[3] =  413.000      Set[4] =  863.000      Set[5] =  22.0000
 Set[6] =  388.000      Set[7] =  307.000      Set[8] =  545.000
 Set[9] =  585.000      Set[10] =  972.000      Set[11] =  417.000
 Set[12] =  572.000      Set[13] =  192.000      Set[14] =  415.000
 Set[15] =  565.000      Set[16] =  814.000      Set[17] =  178.000
 Set[18] =  537.000      Set[19] =  405.000      Set[20] =  765.000
 Set[21] =  380.000      Set[22] =  806.000      Set[23] =  193.000
 Set[24] =  509.000

Data Set B for float type (unsorted):
 Set[0] =  6.28000      Set[1] =  263.000      Set[2] =  75.0000
 Set[3] =  110.000      Set[4] =  514.000      Set[5] =  280.000
 Set[6] =  674.000      Set[7] =  629.000      Set[8] =  864.000
 Set[9] =  806.000      Set[10] =  212.000      Set[11] =  886.000
 Set[12] =  913.000      Set[13] =  519.000      Set[14] =  432.000
 Set[15] =  500.000      Set[16] =  492.000      Set[17] =  569.000
 Set[18] =  791.000      Set[19] =  403.000      Set[20] =  984.000
 Set[21] =  76.0000      Set[22] =  218.000      Set[23] =  882.000
 Set[24] =  333.000

------------------------------------------------------------------------
Data Set A for float type (sorted):
 Set[0] =  3.14000      Set[1] =  22.0000      Set[2] =  114.000
 Set[3] =  178.000      Set[4] =  192.000      Set[5] =  193.000
 Set[6] =  307.000      Set[7] =  380.000      Set[8] =  388.000
 Set[9] =  405.000      Set[10] =  413.000      Set[11] =  415.000
 Set[12] =  417.000      Set[13] =  509.000      Set[14] =  537.000
 Set[15] =  545.000      Set[16] =  565.000      Set[17] =  572.000
 Set[18] =  584.000      Set[19] =  585.000      Set[20] =  765.000
 Set[21] =  806.000      Set[22] =  814.000      Set[23] =  863.000
 Set[24] =  972.000

Statistical Results
   Data Set A Minimum: 3.14000
   Data Set A Maximum: 972.000
```

```
   Data Set A Median: 417.000
   Data Set A Sum: 11544.1
   Data Set A Average: 461.766
   Data Set A Standard Deviation: 254.533

------------------------------------------------------------------------
Data Set B for float type (sorted):
 Set[0] =  6.28000      Set[1] =  75.0000      Set[2] =  76.0000
 Set[3] =  110.000      Set[4] =  212.000      Set[5] =  218.000
 Set[6] =  263.000      Set[7] =  280.000      Set[8] =  333.000
 Set[9] =  403.000      Set[10] =  432.000      Set[11] =  492.000
 Set[12] =  500.000      Set[13] =  514.000      Set[14] =  519.000
 Set[15] =  569.000      Set[16] =  629.000      Set[17] =  674.000
 Set[18] =  791.000      Set[19] =  806.000      Set[20] =  864.000
 Set[21] =  882.000      Set[22] =  886.000      Set[23] =  913.000
 Set[24] =  984.000

Statistical Results
   Data Set B Minimum: 6.28000
   Data Set B Maximum: 984.000
   Data Set B Median: 500.000
   Data Set B Sum: 12431.3
   Data Set B Average: 497.251
   Data Set B Standard Deviation: 291.746

------------------------------------------------------------------------
Summary Results:
   Covariance: 75218.1
   Pearson's Coefficient: 1.01292
   Linear Correlation Coefficient: 1.01292


************************************************************************
Data set - chars

Data for char type (unsorted):
 Set[0] =        y      Set[1] =        u      Set[2] =        a
 Set[3] =        e      Set[4] =        i      Set[5] =        o


Data for char type (sorted):
 Set[0] =        a      Set[1] =        e      Set[2] =        i
 Set[3] =        o      Set[4] =        u      Set[5] =        y


************************************************************************
Data set - chars

Data for string type (unsorted):
 Set[0] =    Black      Set[1] =    Brown      Set[2] =      Red
 Set[3] =   Orange      Set[4] =   Yellow      Set[5] =    Green
 Set[6] =     Blue      Set[7] =   Violet      Set[8] =     Grey
 Set[9] =    White

Data for string type (sorted):
 Set[0] =    Black      Set[1] =     Blue      Set[2] =    Brown
 Set[3] =    Green      Set[4] =     Grey      Set[5] =   Orange
 Set[6] =      Red      Set[7] =   Violet      Set[8] =    White
 Set[9] =   Yellow

************************************************************************
Data set - Long Long

Data for long long type (unsorted):
 Set[0] =      342      Set[1] =      648      Set[2] =      713
 Set[3] =      150      Set[4] =      560      Set[5] =      941
 Set[6] =      762

Data Set Minimum: 300
Data Set Maximum: 1882
Data Set Median: 1296
Data Set Sum: 8232
Data Set Average: 1176
Data Set Standard Deviation: 494
```

```
------------------------------------------------------------------------

Data for long long type (sorted):
 Set[0] =      300      Set[1] =      684      Set[2] =     1120
 Set[3] =     1296      Set[4] =     1426      Set[5] =     1524
 Set[6] =     1882

************************************************************************
Error Testing - Integer Type

Error (constructor), invalid set length.
Error (constructor), invalid set length.
Error (constructor), invalid set length.
Error (constructor), invalid input array.
Error (setDatum), index out or range.
Error (setDatum), index out or range.
Error (getDatum), index out or range.
Error (getDatum), index out or range.
Error (generate), invalid set length.
Error (generate), invalid set length.
Error (generate), invalid set length.
Error (coVariance), invalid data set sizes.
Error (coVariance), invalid data set sizes.
Error (perasonsCoefficient), invalid data set sizes.
Error (perasonsCoefficient), invalid data set sizes.
Error (linearCorrelationCoefficient), invalid data set sizes.
Error (linearCorrelationCoefficient), invalid data set sizes.

------------------------------------------------------------------------

Data for int type (sorted):
 Set[0] =       17      Set[1] =      339      Set[2] =      591
 Set[3] =      736      Set[4] =      824

************************************************************************
Data set - Shorts

Enter Count (5-9999): seven

Error, count out of range.
Enter Count (5-9999): @3

Error, count out of range.
Enter Count (5-9999): 4

Error (read), count Value 4 not between 5 and 9999.
Enter Count (5-9999): 10000

Sorry, too many errors.
ed-vm%
```