**New York University**

**School of Continuing Studies**

**Information Technologies Institute**

# C++ Part I

**Take Home Midterm Exam**

**Fall 2014 Yedidiah Solowiejczyk**

**INFO1\_CE9264-01 Wed 6:00 – 9:30 PM**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Write C++ programs for each of the following questions. The source code and execution output should be provided separately. The exam is due **Nov 12, 2014**.

1. Write a program that calculates the **arithmetic mean value** and **standard deviation** of the following array of numbers.

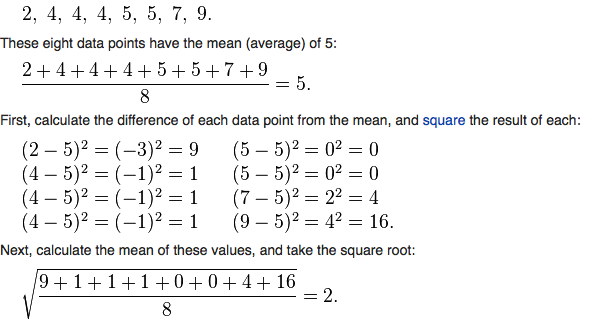
**float Array[100] = {1.0,2.0,3.0,4.0,…..99.00, 100.0}**

**Int Fib[20] = {0,1,1,2,3,5……….}**

The **arithmetic mean value** of an array is defined as:



whereas the **standard deviation(SD)** is defined as follows:



* The program should generate the data for each type of array.
* The program should have **overloaded function** that are declared as follows:
* **float Mean(int array[ ], int Size);**
* **float Mean(float array[ ], int Size = 10);** //default value
* **float SD(int array[ ], int Size);**
* **float SD(float array[ ], int Size = 10);** //default value

You should provide answers for the following cases:

* calculate the mean of Array[100]
* calculate the SD of Array[100]
* calculate the mean for Fib[20]
* calculate the SD for Fib[20]
* calculate the SD for Array[10] – first 10 elements of array
* recognize that the SD function invokes the mean function.

1. The power function, int **Pow(int X, int N),**  returns the value of X raised to the Nth power.

**XN = X\*X\*X\*…..X (N times)**

(a) Write a **recursive** function **Pow(int X, int N)** that will calculate the output value for the following cases:

Int Result = Pow(10,3);

Int Result = Pow(2,10);

Int Result = Pow(1024,10);

As part of getting a better idea on how the stack works, you should declare a temporary int\* inside Pow( ) and print its address on each visit to the recursive function Pow( ).

(b) As an alternative solution write a function that **POW**(int X, Int N) that is not

recursive but recognizes that X1024  = X512\*X512 = (X256\*X256) (X128\*X128)(X128\*X128) …… etc

* Derive the same results as in part (a)

**--------------------------------------------------------------------------------**

1. **A class *Point* is defined by the following attributes and member functions:**

* **int x, y; // location on an x-y coordinate system**
* **default constructor(write message – dc)**
* **parameterized constructor(write message – pc)**
* **destructor (write message – “going out of scope”)**
* **copy constructor (write message – cc)**
* **void reset\_location(int value\_x, int value\_y)**
* **void move(int delta\_x, int delta\_y)**
* **int getX( )**
* **int getX( )const**
* **int getY( )**
* **int getY( )const**

1. **create** 
   1. **create Point P1,P2(3,4)**
   2. **create const P3(10,4)**
   3. **move P2 by 5 units in x direction and 10 unites in y direction**
   4. **get the X value of P2 and Y value of P3 and demonstrate (via internal message) that each object has gravitated to correct function**
   5. **create an array of Points Point\_Array[5] and initialize them to form a straight line**
   6. **print the values of the array Point\_Array**
   7. **demonstrate void Reverse(Point & P1, Point P2); //pass Point objects by reference and value that prints its x,y coordinates**

**Develop the code for the above class, debug and test.**

**Good Luck…….!!!!**

**Feel free to email me if you have any problems or questions. I encourage everyone to start looking at the problems this week.**

**ysolowie@yahoo.com**