

Lab 11

ARRAYS

Dr. John P. Abraham

If we want to write a program to find the standard deviation of a set of test scores we need to have a variable that could hold multiple integers instead of assigning an identifier for each score. An array is the answer to our need. An array is used to store and process a **collection of data of the same type**. Let me emphasize that an array can hold **multiple values of the same type**; it cannot hold values of different types (we saw that a struct can do that). An array may also be called a list.

First let us talk about declaring an array. Any program that we might write should be general enough to apply to different situations; the program we are about to write should not only let us calculate the standard deviation for exam #1 but also all subsequent exams. Therefore, we have to decide what is the maximum number of scores we will have. For now, we will choose 90. This is a limitation of arrays; we are limited to this number once we code it in. To change it we will have to change the code and recompile. That is why we call an array a **static list**. Back to declaring the array. To declare an array of 90 scores declare it like any other integer variable, but put a [90] in the square brackets: **int scores[90]**.

Just because we declared an array for 90 elements, there is no need to enter all 90 scores. We can enter any number of scores up to 90. Let us write a program to accept some scores into an array (terminate the entry when a negative number is entered). We will have to keep track of the number scores entered. Program Six_1 uses a global variable for the array to keep the example simple. Program Six_2 shows passing the array as an argument. Please pay special attention to the fact that the n is decreased by one to get the actual number of valid scores entered. The last grade (negative score), although counted, is not a valid score.

Program 11-1

```
/******  
Accept some scores into an array.  
Terminate Entry when a negative number is entered.  
Keep track of number of scores entered.  
Teaching objective - Data entry into an array.  
By Dr. John Abraham  
Created for 1370 students  
*****/  
  
#include <iostream>  
using namespace std;  
  
int scores[90]; //array is declared globally for this example.  
int getscores (void);
```

```

int main ()
{
    int n, i; // n for number of scores. i is a counter.
    n = getscores(); //go get the scores and how many
    for (i=1; i<=n; i++) cout<<scores[i]<<endl; //show scores.
    return(0);
}

int getscores()
{
    int n=1;
    cout << "ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY
ENTERING A NEG NUMBER!\n";
    cout << "Enter score# " << n << " ";
    cin >> scores[n];
    while (scores[n] >= 0)
    {
        n++;
        cout << "Enter score# " << n <<" ";
        cin >> scores[n];
    }
    return n-1; //n-1 actual scores read.
}

```

Program Run 11-1.

```

ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY ENTERING A
NEG NUMBER!
Enter score# 1 78
Enter score# 2 91
Enter score# 3 88
Enter score# 4 75
Enter score# 5 60
Enter score# 6 88
Enter score# 7 59
Enter score# 8 99
Enter score# 9 78
Enter score# 10 -1
78
91
88
75
60
88

```

59

99

78

Press any key to continue

Passing array as a parameter is not all that difficult. First, include a prototype which indicates that the parameter passed is an array, without actually showing the number of elements in the array - just put []. The function heading also has similar declaration. The variable declaration however, should show the dimension of the array. Please compare Program 11-1 to Program 11-2 to see the differences.

Program 11_2.

```
/******  
Accept some scores into an array.  
Terminate Entry when a negative number is entered.  
Keep track of number of scores entered.  
Teaching objective - Pass Array as a parameter.  
By Dr. John Abraham  
Created for 1370 students  
*****/  
  
#include <iostream>  
using namespace std;  
  
int getscores (int scores[]);      //prototype for passing an array  
  
int main ()  
{  
    int scores[90];                //array of 90 integers named  
scores  
    int n, i;                      // n for number of scores.  
i is a counter.  
    n = getscores(scores);        //go get the scores and how many  
    for (i=1; i<=n; i++) cout<<scores[i]<<endl;    //show scores.  
    return(0);  
}  
  
int getscores(int scores[])  
{  
    int n=1;  
    cout << "ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY  
ENTERING A NEG NUMBER!\n";  
    cout << "Enter score# " << n << " ";  
    cin >> scores[n];  
    while (scores[n] >= 0)  
    {  
        n++;  
    }  
}
```

```
        cout << "Enter score# " << n <<" ";
        cin >> scores[n];
    }
    return n-1; //n-1 actual scores read.
}
```

Program Run 11-2.

```
ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY ENTERING A
NEG NUMBER!
Enter score# 1 88
Enter score# 2 77
Enter score# 3 76
Enter score# 4 -1
88
77
76
Press any key to continue
```

It is important to note that the array parameter passed to the function is neither a **call-by-value**, or a **call-by-reference**, but a new kind of parameter known as an **array parameter**. It follows that when passing an array neither a copy of the entire array is passed nor every memory location of entire array are passed, instead the memory location of the first element of the array is passed. Therefore, it is not necessary to pass the size of the array, just include square brackets with nothing in it. Both the calling module and called module calculate the memory location of a particular element using abase and offset (calculated by index). More about addressing modes will be taught in future courses.

In developing this chapter I will use the standard deviation example. So let me explain what it is. In simple terms the standard deviation is a weighted average of differences of all the scores from the mean. In order to calculate it we follow the following steps:

1. Find the mean (average) of all the scores.
2. Find the difference of each score from the mean.
3. Square the differences (item #2).
4. Find the sum of all squared differences (item #3).
5. Find the variance by dividing the sum (item #4) by the number of scores minus one.
6. Find the square root of the variance (item #5).

1. Find the mean of the scores entered.

Program 11_3.

```
/******  
Accept some scores into an array.  
Terminate Entry when a negative number is entered.  
Keep track of number of scores entered.  
Teaching objective - Pass Array as a parameter.  
By Dr. John Abraham  
Created for 1370 students  
*****/  
  
#include <iostream>  
using namespace std;  
  
int getscores (int scores[]);  
float calcMean (int scores[], int);  
void display(int, int scores[], float);  
  
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
int main ()  
{  
    int scores[90];  
    int n; // n for number of scores.  
    float mean;  
    n = getscores(scores); //go get the scores and how many  
    mean = calcMean(scores, n);  
    display(n,scores,mean);  
    return(0);  
}  
  
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
int getscores(int scores[])  
{  
    int n=1;  
    cout << "ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY  
ENTERING A NEG NUMBER!\n";  
    cout << "Enter score# " << n << " ";  
    cin >> scores[n];  
    while (scores[n] >= 0)  
    {  
        n++;  
        cout << "Enter score# " << n <<" ";  
        cin >> scores[n];  
    }  
  
    return n-1; //n-1 actual scores read.  
}  
  
//ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff  
float calcMean(int scores[], int n)  
{  
    int sum=0;  
    int i; //use for counter  
    float m; //local variable for mean  
    for (i=1; i <=n; i++) sum += scores[i]; //add all scores  
    m = float(float(sum)/n);  
}
```

```

        return m;
    }

//ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
void display (int n, int scores[], float mean)
{
    int i;
    cout << n<< " Scores were entered. They are: \n";
    for (i=1; i<=n; i++) cout<<scores[i]<<endl;    //show scores.
    cout << "The mean is: "<<mean <<endl;
}

```

Program Run 11 3.

ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY ENTERING A NEG NUMBER!

```

Enter score# 1 88
Enter score# 2 77
Enter score# 3 83
Enter score# 4 85
Enter score# 5 96
Enter score# 6 68
Enter score# 7 -1
6 Scores were entered. They are:
88
77
83
85
96
68
The mean is: 82.8333
Press any key to continue

```

Standard deviation program continued.

- 2. Find difference of each score from the mean.**
- 3. Square the differences.**
- 4. Add the squared differences.**
- 5. Find the variance by dividing the sum by the number of scores minus one.**
- 6. Find the square root of the variance.**

Program 11 4.

```

/*****
Accept some scores into an array.
Terminate Entry when a negative number is entered.
Keep track of number of scores entered.
Find difference of each score from the Mean.

```



```

        float m; //local variable for mean
        for (i=1; i <=n; i++) sum += scores[i]; //add all scores
        m = float(float(sum)/n);
        return m;
    }

//ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff

float sumSquares (int n, int scores[], float diff[],
                 float diffSq[],float mean)

    {
        int i; float ss=0.0;
        for (i=1; i <=n; i++)
        {
            diff[i] = scores[i]-mean;
            diffSq[i] = diff[i] * diff[i];
            ss += diffSq[i];
        }
        return ss;
    }

//ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff

float stdDeviation(int n, float sumSq)

    {
        float variance;
        variance = sumSq/(n-1);
        return (float(sqrt(variance)));
    }

//ffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff

void Table(int n, int scores[], float diff[],
           float diffSq[], float mean, float stDev)
    {
        int i; float ss=0;
        cout << "\nStadard Deviation for " << n<<" Scores. Mean: "<<mean
        <<endl;
        cout <<"-----\n";
        cout <<setw(10) <<"Score"<<setw(10) <<"Dev" <<setw(10) << "Dev Sq"
            <<" Sum of dev2)\n";
        cout <<"-----\n";
        cout << setw(10) << setiosflags(ios::fixed);

        for (i = 1; i<=n; i++)
        {
            ss += diffSq[i];
            cout << setw(10) << setprecision(2) << scores[i]
                << setw(10) << setprecision(2) << diff[i]
                << setw(10) << setprecision(2) << diffSq[i]
                << setw(10) << setprecision(2) << ss << endl;
        }
        cout <<"-----\n";
    }

```



```
cout <<"Standard Deviation " << setprecision(2) <<stDev <<endl;
}
```

Program Run 11-4

```

c:\c-2005\stddev\debug\stdDev.exe
ENTER A SCORE AND PRESS ENTER. YOU QUIT ANY TIME BY ENTERING A NEG NUMBER!
Enter score# 1 88
Enter score# 2 89
Enter score# 3 87
Enter score# 4 85
Enter score# 5 56
Enter score# 6 99
Enter score# 7 78
Enter score# 8 89
Enter score# 9 85
Enter score# 10 -1

Stadard Deviation for 9 Scores. Mean: 84
-----
Score      Dev    Dev Sq  Sum of dev2>
-----
88         4.00   16.00   16.00
89         5.00   25.00   41.00
87         3.00    9.00   50.00
85         1.00    1.00   51.00
56        -28.00  784.00  835.00
99        15.00  225.00  1060.00
78        -6.00   36.00  1096.00
89         5.00   25.00  1121.00
85         1.00    1.00  1122.00
-----
Standard Deviation 11.84

```

The program 11-4 is modified to read the scores from a file. A file must be created and placed in the same subdirectory where the source code is saved. You can create the file using notepad. Make sure to save the file with the name “scores1.dta”. In order to do that you must change the file type to “all files” instead of default text files. Alternatively you can save the file as a text file using any name, but make sure to change the name of the file in your source code (example “myfile.txt”).

```

/*****
Read scores from a file into an array.
Keep track of number of scores entered.
Find difference of each score from the Mean.
Square the differences and add the squares.
find stadard deviation
create a table
Teaching objective - Pass Array as a parameter.
By Dr. John Abraham
Created for 1370 students
*****/

#include <iostream>
#include <fstream>
#include <iomanip>
#include <cmath> //for square root function.

using namespace std;

```

```

int getscores (int scores[]);
float calcMean (int scores[], int);
float sumSquares (int, int scores[], float diff[], float
diffSq[],float);
float stdDeviation(int, float);
void Table(int, int scores[], float diff[], float diffSq[], float,
float);
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
int main ()
{
    int scores[90];float diff[90], diffSq[90];
    int n; // n for number of scores.
    float mean, sumSq, stDev;
    n = getscores(scores); //go get the scores and how many
    mean = calcMean(scores, n);
    sumSq = sumSquares (n,scores,diff,diffSq,mean);
    stDev = stdDeviation(n, sumSq);
    Table(n, scores, diff,diffSq, mean, stDev);
    cin >>n;
    return(0);
}

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

int getscores(int scores[])
{
    ifstream infile;
    infile.open("scores1.dta");
    if (!infile) cout << "Data file does not exist!";
    int n=0;
    while(!infile.eof())

        {
            n++;
            infile >> scores[n];
        }
    return n; //n-1 actual scores read.
}

//fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
float calcMean(int scores[], int n)
{
    int sum=0;
    int i; //use for counter
    float m; //local variable for mean
    for (i=1; i <=n; i++) sum += scores[i]; //add all scores
    m = float(float(sum)/n);
    return m;
}

//fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff

float sumSquares (int n, int scores[], float diff[],
float diffSq[],float mean)

```

```

    {
        int i; float ss=0.0;
        for (i=1; i <=n; i++)
        {
            diff[i] = scores[i]-mean;
            diffSq[i] = diff[i] * diff[i];
            ss += diffSq[i];
        }
        return ss;
    }

//fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
float stdDeviation(int n, float sumSq)

{
    float variance;
    variance = sumSq/(n-1);
    return (float(sqrt(variance)));
}

//fffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff

void Table(int n, int scores[], float diff[],
           float diffSq[], float mean, float stDev)
{
    int i; float ss=0;
    cout << "\nStandard Deviation for " << n<<" Scores. Mean: "<<mean
    <<endl;
    cout <<"-----\n";
    cout <<setw(10) <<"Score"<<setw(10) <<"Dev" <<setw(10) << "Dev Sq"
        <<" Sum of dev2)\n";
    cout <<"-----\n";
    cout << setwiosflags(ios::fixed);

    for (i = 1; i<=n; i++)
    {
        ss += diffSq[i];
        cout << setw(10) << setprecision(2) << scores[i]
            << setw(10) << setprecision(2) << diff[i]
            << setw(10) << setprecision(2) << diffSq[i]
            << setw(10) << setprecision(2) << ss << endl;
    }
    cout <<"-----\n";
    cout <<"Standard Deviation " << setprecision(2) <<stDev <<endl;
}

```

Assignment:

Write a program to convert a number between 0 and 20000 (exclusive) into words. The program must use arrays to hold words such as “one, two, three, etc.” Here is a sample run.

```
"z:\abrahamfiles\netc\numberconvert\debug\numberconvert.exe"  
This program will convert a number to words.  
Enter a number less than 20,000 : 18975  
  
Number      :18975  
In Words    :eighteen thousand nine hundred seventy-five  
  
Press any key to continue
```