Solution Review: Calculate Mean and Standard Deviation from Array

Let's go over the solution review of the challenge given in the previous lesson.

We'll cover the following

- Solution
- Explanation

(/learn)

Solution#

Press the **RUN** button and see the output!

```
    Array[i] = rand() % 100;
  }

  //Prints dynamic array
  cout << "Elements of array: ";
  for(int i = 0 ; i < size; i++){
    cout << Array[i] << " ";
  }
  cout << endl;

  //Calculate mean and print
  float sum = 0;
  for(int i = 0 ; i < size; i++){
    sum += Array[i];
  }
  float mean = sum/size ;
  cout << "Mean: " << mean << endl;

  //Calculate standard deviation and print
  float stdDev = 0;
```

```cpp
  for(int i = 0 ; i < size; i++){
    stdDev += pow(Array[i] - mean, 2);
  }
  stdDev = sqrt(stdDev / size);
  cout << "Standard Deviation: " << stdDev << endl;
  // Deletes a memory allocated to dynamic array
  delete[] Array;
}
```

Output 1.43s

```
Elements of array: 83 86 77 15 93 35 86 92 49 21
Mean: 63.7
Standard Deviation: 29.0484
```

Explanation#

- First, we allocated the dynamic array of size 10.
- Then, we initialized the array with random numbers using the `rand()` method.
- The elements of the array are then printed using a for loop.
- Then we calculated the `mean` by calculating the `sum` and then dividing it by `size`.
- Afterwards, the standard deviation is calculated by finding the average of the squared differences from the mean and then taking a square root of it.
- In the end, the memory of the dynamic array was deallocated by using `delete` keyword.

← Back

Next →