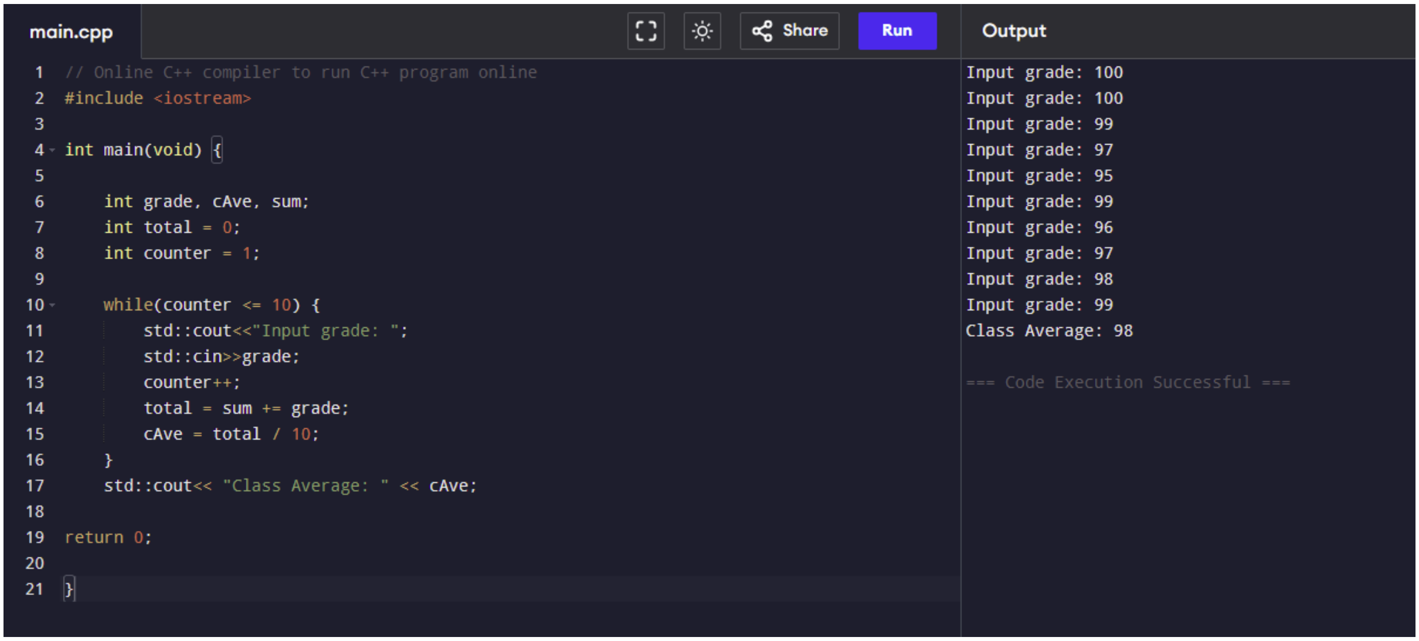
Activity No. 2.2	
Control Structures	
Course Code: CPE007	**Program:** Computer Engineering
Course Title: Programming Logic and Design	**Date Performed: 08/07/25**
Section: CPE11S1	**Date Submitted: 12/08/25**
Name(s): James Daniel M. Verano	**Instructor: Engr. Jimlord M. Quejado**
6. Output	

Exercise 1: *Counter- Controlled Repetition. A class of ten students took a quiz. The grades (integers in the range of 0 to 100) for this quiz are available to you. Determine the class average on the quiz. Put your answer in the output section of the activity template. Ensure that the screen shot of the code and the output are readable.*

*Using the following **pseudocode** the program can be as follows:*

```
Set total to zero
Set grade counter to one
While grade counter is less than or equal to ten
Input the next grade
Add the grade into the total
Add one to the grade counter
Set the class average to the total divided by ten
Print the class average
```

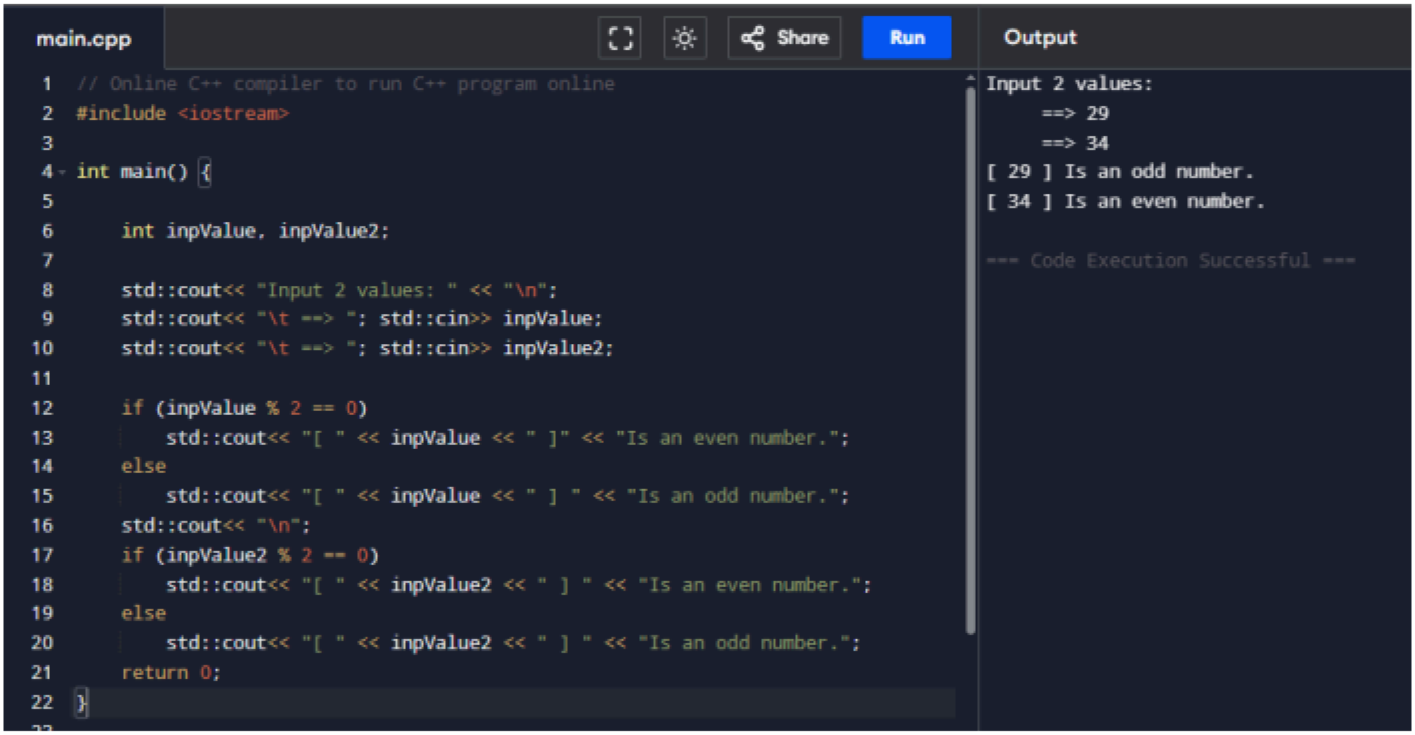


```cpp
// Online C++ compiler to run C++ program online
#include <iostream>

int main(void) {

    int grade, cAve, sum;
    int total = 0;
    int counter = 1;

    while(counter <= 10) {
        std::cout<<"Input grade: ";
        std::cin>>grade;
        counter++;
        total = sum += grade;
        cAve = total / 10;
    }
    std::cout<< "Class Average: " << cAve;

return 0;

}
```

```
Input grade: 100
Input grade: 100
Input grade: 99
Input grade: 97
Input grade: 95
Input grade: 99
Input grade: 96
Input grade: 97
Input grade: 98
Input grade: 99
Class Average: 98

=== Code Execution Successful ===
```

7. Supplementary Activity

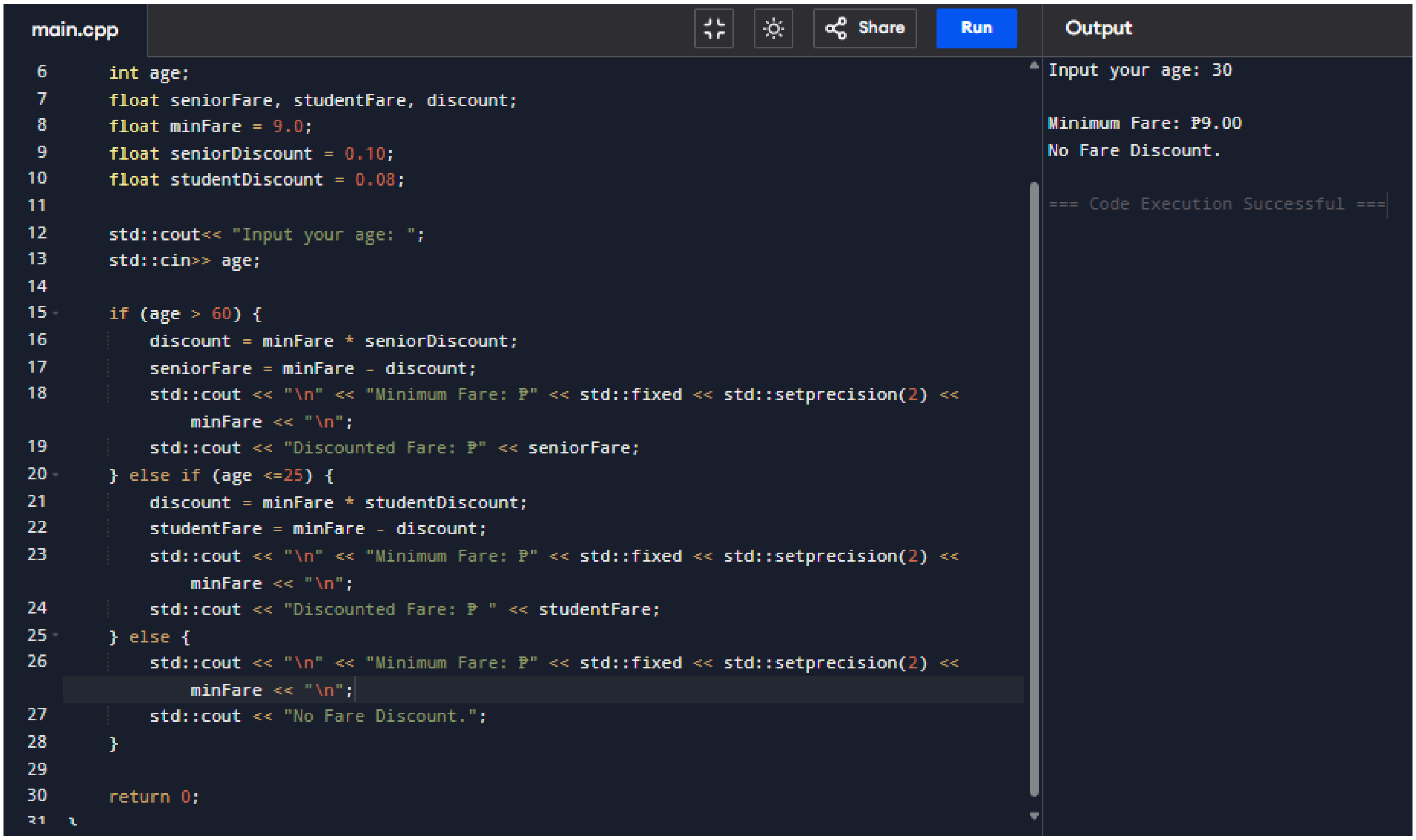
1. *Using conditional statements (if-else statements), write a program that asks a user for a number and prints out if it is an even or an odd number.*



```cpp
// Online C++ compiler to run C++ program online
#include <iostream>

int main() {

    int inpValue, inpValue2;

    std::cout<< "Input 2 values: " << "\n";
    std::cout<< "\t ==> "; std::cin>> inpValue;
    std::cout<< "\t ==> "; std::cin>> inpValue2;

    if (inpValue % 2 == 0)
        std::cout<< "[ " << inpValue << " ]" << "Is an even number.";
    else
        std::cout<< "[ " << inpValue << " ] " << "Is an odd number.";
    std::cout<< "\n";
    if (inpValue2 % 2 == 0)
        std::cout<< "[ " << inpValue2 << " ] " << "Is an even number.";
    else
        std::cout<< "[ " << inpValue2 << " ] " << "Is an odd number.";
    return 0;
}
```

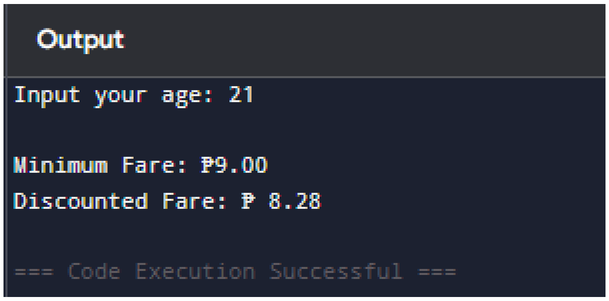
```
Input 2 values:
        ==> 29
        ==> 34
[ 29 ] Is an odd number.
[ 34 ] Is an even number.

=== Code Execution Successful ===
```

2. *Using conditional statements, write a program that computes for 10 percent fare discount of a senior citizen and 8 percent fare discount of a student. There will be no discount if not a senior citizen and not a student. The user will be asked to enter age. The minimum fare is 9 pesos.*

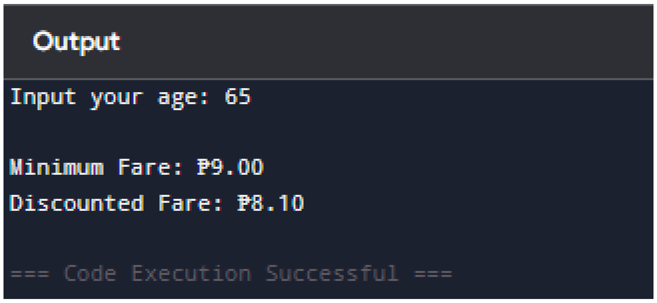


```
    int age;
    float seniorFare, studentFare, discount;
    float minFare = 9.0;
    float seniorDiscount = 0.10;
    float studentDiscount = 0.08;

    std::cout<< "Input your age: ";
    std::cin>> age;

    if (age > 60) {
        discount = minFare * seniorDiscount;
        seniorFare = minFare - discount;
        std::cout << "\n" << "Minimum Fare: ₱" << std::fixed << std::setprecision(2) <<
            minFare << "\n";
        std::cout << "Discounted Fare: ₱" << seniorFare;
    } else if (age <=25) {
        discount = minFare * studentDiscount;
        studentFare = minFare - discount;
        std::cout << "\n" << "Minimum Fare: ₱" << std::fixed << std::setprecision(2) <<
            minFare << "\n";
        std::cout << "Discounted Fare: ₱ " << studentFare;
    } else {
        std::cout << "\n" << "Minimum Fare: ₱" << std::fixed << std::setprecision(2) <<
            minFare << "\n";
        std::cout << "No Fare Discount.";
    }

    return 0;
}
```

```
Input your age: 30

Minimum Fare: ₱9.00
No Fare Discount.

=== Code Execution Successful ===
```



```
Input your age: 21

Minimum Fare: ₱9.00
Discounted Fare: ₱ 8.28

=== Code Execution Successful ===
```

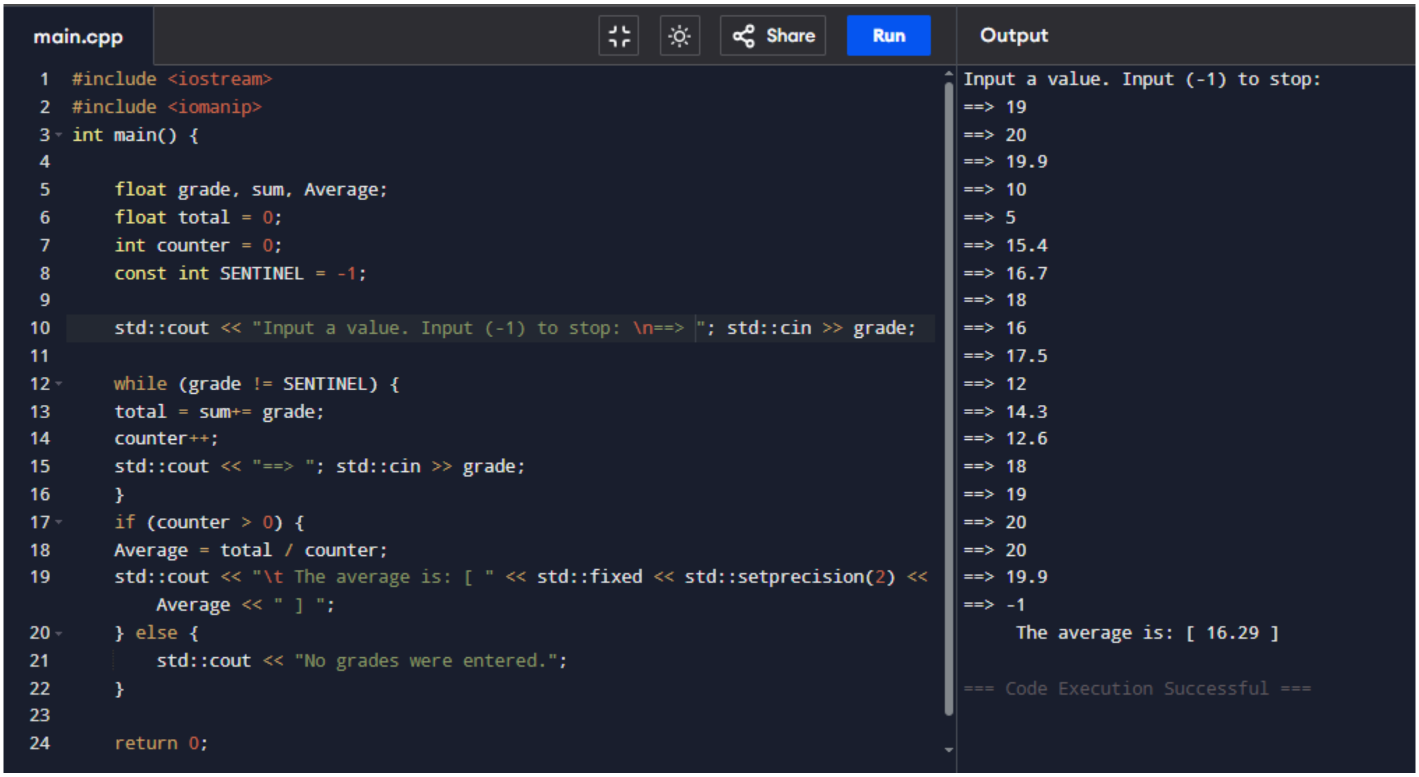


```
Input your age: 65

Minimum Fare: ₱9.00
Discounted Fare: ₱8.10

=== Code Execution Successful ===
```

3. **Case Study**: Sentinel Controlled Repetition. Given the following pseudocode, create a program that will implement a sentinel controlled repetition. For example, you can use (-1) as the sentinel value. You can use Problem 1 as your reference.

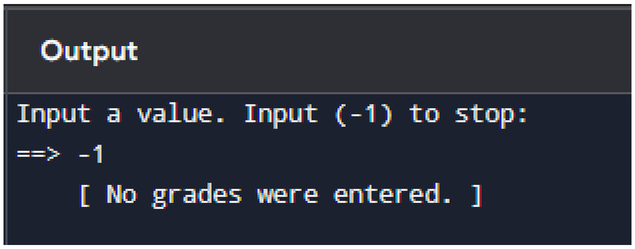
Pseudocode:

```
Initialize total to zero
Initialize counter to zero


Input the first grade
While the user has not as yet entered the sentinel
Add this grade into the running total
Add one to the grade counter
Input the next grade (possibly the sentinel)
If the counter is not equal to zero
Set the average to the total divided by the counter
Print the average
else
Print “No grades were entered”
```



```cpp
#include <iostream>
#include <iomanip>
int main() {

    float grade, sum, Average;
    float total = 0;
    int counter = 0;
    const int SENTINEL = -1;

    std::cout << "Input a value. Input (-1) to stop: \n==> "; std::cin >> grade;

    while (grade != SENTINEL) {
    total = sum+= grade;
    counter++;
    std::cout << "==> "; std::cin >> grade;
    }
    if (counter > 0) {
    Average = total / counter;
    std::cout << "\t The average is: [ " << std::fixed << std::setprecision(2) <<
        Average << " ] ";
    } else {
        std::cout << "No grades were entered.";
    }

    return 0;
```

```
Input a value. Input (-1) to stop:
==> 19
==> 20
==> 19.9
==> 10
==> 5
==> 15.4
==> 16.7
==> 18
==> 16
==> 17.5
==> 12
==> 14.3
==> 12.6
==> 18
==> 19
==> 20
==> 20
==> 19.9
==> -1
     The average is: [ 16.29 ]

=== Code Execution Successful ===
```



```
Input a value. Input (-1) to stop:
==> -1
    [ No grades were entered. ]
```

8. Conclusion

This activity has served me to explore the functionality of various control structures in programming such as looping, definite repetition, and Sentinel controlled repetition. With the given exercise, I was able to learn and enhance my programming skills, and the understanding of the basic programming logic and design subj.

9. Assessment Rubric