

Activity No. 2.2

Hands-on Activity 2.2: Control Structures (Part 1)

Course Code: CPE007	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: August 7, 2025
Section: CPE11S1	Date Submitted: August 11, 2025
Name(s): Mendoza, Nathaniel B.	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.

main.cpp	Run	Output
<pre>1 // Exercise 1, MENDOZA, NATHANIE B. 2 // Using the following Pseudocode to program. 3 #include <iostream> 4 int main() { 5 int total = 0; // Setting Total of 0 6 int grade; // Grade 7 int gradeCounter = 1; // Setting Grade counter to 1 8 while (gradeCounter <= 10) //Grade Counter till less than or equal to 10 9 { 10 std::cout << "Enter grade for student " << gradeCounter << ":"; 11 std::cin >> grade; 12 13 total += grade; // Adding the grade of 1 student in the total 14 gradeCounter++; // Next student 15 } 16 17 double average = static_cast<double>(total) / 10; // All Scores of Student then divide by 10 to get Average 18 std::cout << "\nClass average is: " << average << std::endl; 19 20 return 0; 21 }</pre>	  	Enter grade for student 1: 90 Enter grade for student 2: 98 Enter grade for student 3: 77 Enter grade for student 4: 99 Enter grade for student 5: 66 Enter grade for student 6: 8 Enter grade for student 7: 7 Enter grade for student 8: 66 Enter grade for student 9: 100 Enter grade for student 10: 99 Class average is: 71 == 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.

<pre> 1 // Exercise 1, MENDOZA, NATHANIE B. 2 // Odd or Even using if and else statement 3 #include <iostream> 4 int main() 5 { 6 int number; 7 std::cout << "Enter a number: "; 8 std::cin >> number; 9 if (number % 2 == 0) //Setting number that is not divisible by 2 as 10 Odd 11 { 12 std::cout << number << " is even." << std::endl; 13 } else 14 { 15 std::cout << number << " is odd." << std::endl; 16 } 17 return 0; </pre>	Run	<p>Output</p> <pre> Enter a number: 3 3 is odd. ==== Code Execution Successful === </pre>
---	---	--

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.

<pre> 1 // Exercise 1, MENDOZA, NATHANIE B. 2 // Odd or Even using if and else statement 3 #include <iostream> 4 int main() 5 { 6 int age; 7 double fare = 9.0; // Minimum Fare is 9 Pesos. 8 std::cout << "Enter age: "; 9 std::cin >> age; 10 if (age >= 60) 11 { 12 // If Age is >60 set as Senior Citizen 13 fare -= fare * 0.10; // 10% discount if Senior Citizen 14 std::cout << "Senior Citizen Discount applied." << std::endl 15 } 16 else if (age >= 5 && age <= 22) //Kindergarten to College 17 { 18 fare -= fare * 0.08; // 8% discount if Student 19 std::cout << "Student Discount applied." << std::endl; 20 } 21 else //No discount if age is above 23 to 59 22 { 23 std::cout << "No discount applied." << std::endl; 24 } 25 std::cout << "Fare: " << fare << " pesos" << std::endl; 26 return 0; 27 </pre>	Run	<p>Output</p> <pre> Enter age: 44 No discount applied. Fare: 9 pesos ==== Code Execution Successful === </pre>
--	---	---

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.

The screenshot shows a code editor interface with a tab for "main.cpp". The code itself is a C++ program that calculates the average of student grades using sentinel-controlled repetition. It includes comments, variable declarations, a loop for input, and an if-statement to handle the case where no grades were entered. The output window shows the execution of the program, prompting for grades and displaying the class average. The run was successful.

```
1 //MENDOZA, NATHANIEL B CPE11S1
2 //SENTINEL CONTROLLED REPITITION
3 #include <iostream>
4 int main() {
5     int total = 0;
6     int counter = 0;
7     int grade;
8     std::cout << "Enter grades (-1 to stop): ";
9     std::cin >> grade;
10    while (grade != -1) {
11        total += grade;
12        counter++;
13        std::cout << "Enter grade (-1 to stop): ";
14        std::cin >> grade;
15    }
16    if (counter != 0) {
17        double average = static_cast<double>(total) / counter;
18        std::cout << "\nClass average is: " << average << std::endl;
19    } else {
20        std::cout << "No grades were entered." << std::endl;
21    }
22    return 0;
23 }
```

Output:

```
Enter grades (-1 to stop): 99
Enter grade (-1 to stop): 65
Enter grade (-1 to stop): 98
Enter grade (-1 to stop): 68
Enter grade (-1 to stop): 75
Enter grade (-1 to stop): -1
Class average is: 81
== Code Execution Successful ==
```

8. Conclusion

Through these activities, I was able to practice using conditional statements and loops to create programs that respond to different situations such as by entering student grades, making an odd or even number and the program will define and lastly define the age and which discount will be given. The counter-controlled repetition showed me how useful it is when the exact number of inputs is known, while the sentinel-controlled repetition taught me how to handle cases where the number of inputs is uncertain. Overall, these exercises helped me understand how selection and repetition structures work together to make programs more dynamic, and practical for real-life applications.

9. Assessment Rubric