

Activity No. <n>	
<Replace with Title>	
Course Code: CPE010	Program: Computer Engineering
Course Title: Data Structures and Algorithms	Date Performed: 08/07/2025
Section: CPE11S1	Date Submitted: 08/12/25
Name(s): Nieles, Cenndy M.	Instructor: Engr. Jimlord Quejado

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*

```
#include <iostream>

int total = 0
int grade_counter = 1;
int grade; double average; // this will be able to display decimals

while (grade_counter <= 10) { // this will create a loop
    std::cout << "Enter Grade " << grade_counter << ": ";
    std::cin >> grade;

    total += grade;
    grade_counter++;
}

average = static_cast<double>(total) / 10;
std::cout << "\nClass average is: " << average << std::endl;

return 0;
```

Results:

The screenshot shows a code editor interface with a dark theme. On the left, the code file 'main.cpp' contains C++ code for calculating the average of 10 grades. On the right, the 'Output' panel shows the execution results: ten grade inputs followed by the calculated average and a success message.

```
main.cpp
1 #include <iostream>
2
3 int main() {
4
5     int total = 0;
6     int grade_counter = 1;
7     int grade;
8     double average; // this will be able to display decimals
9
10    while (grade_counter <= 10) { // this will create a loop
11        std::cout << "Enter Grade " << grade_counter << ": ";
12        std::cin >> grade;
13
14        total = total + grade;
15        grade_counter = grade_counter + 1;
16    }
17
18    average = static_cast<double>(total) / 10;
19    std::cout << "\nClass average is: " << average << std::endl;
20
21    return 0;
22 }
```

Output

```
Enter Grade 1: 91
Enter Grade 2: 92
Enter Grade 3: 93
Enter Grade 4: 94
Enter Grade 5: 95
Enter Grade 6: 96
Enter Grade 7: 97
Enter Grade 8: 98
Enter Grade 9: 99
Enter Grade 10: 100
Class average is: 95.5
*** Code Execution Successful ***
```

7. Supplementary Activity

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.

```
#include <iostream>

int main() {

    int value;

    std::cout << "Enter Value: ";
    std::cin >> value;

    if (value % 2 == 0)
        std::cout << value << " is an even value." << std::endl;

    else
        std::cout << value << " is an odd value." << std::endl;

    return 0;
}
```

RESULT

Programiz C++ Online Compiler

Programiz PRO >

main.cpp	Run	Output
1 #include <iostream> 2 3 int main() { 4 int value; 5 std::cout << "Enter Value: "; 6 std::cin >> value; 7 8 if (value % 2 == 0) 9 std::cout << value << " is an even value." << std::endl; 10 else 11 std::cout << value << " is an odd value." << std::endl; 12 13 return 0; 14 } 15 16 17 18 19	Enter Value: 100 100 is an even value. == Code Execution Successful ==	

ODD VALUE

Programiz C++ Online Compiler

Programiz PRO >

main.cpp	Run	Output
3 int main() { 4 5 int value; 6 7 std::cout << "Enter Value: "; 8 std::cin >> value; 9 10 if (value % 2 == 0) 11 std::cout << value << " is an even value." << std::endl; 12 else 13 std::cout << value << " is an odd value." << std::endl; 14 15 return 0; 16 }	Enter Value: 99 99 is an odd value. == Code Execution Successful ==	

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.

CODE:

```
#include  
  
int main() {  
  
int age;  
  
double fair = 9.0; double fairfinal;  
  
std::cout << "Enter Your Age: ";  
  
std::cin >> age;
```

```

if (age >= 60) { // 60 years old is considered being a senior citizen (Philippines).

fairfinal = fair - (fair * 0.10);

std::cout<<"Senior Discount Fair." <<fairfinal<< "Pesos"<<std::endl;

}

else if ( age <=25) { // anyone could be still studying at a later age but im going with this

fairfinal = fair - (fair * 0.08);

std::cout<< "Student Discount Fair: " << fairfinal<< " Pesos"<< std::endl;

}

else { fairfinal = fair; // no discount

std::cout<< "Regular Fair: " << fairfinal<< " Pesos"<< std::endl;

}

return 0;

```

The screenshot shows a code editor interface with the following details:

- Code Area (main.cpp):**

```

1 #include <iostream>
2
3 int main()
4 {
5     int age;
6     double fair = 9.0;
7     double fairfinal;
8
9     std::cout << "Enter Your Age: ";
10    std::cin >> age;
11
12    if (age >= 60) { // 60 years old is considered being a senior citizen (Philippines).
13        fairfinal = fair - (fair * 0.10);
14        std::cout << "Senior Discount Fair: " << fairfinal << " Pesos" << std::endl;
15    }
16    else if (age <= 25) { // anyone could be still studying at a later age but Im going with
17        // this
18        fairfinal = fair - (fair * 0.08);
19        std::cout << "Student Discount Fair: " << fairfinal << " Pesos" << std::endl;
20    }
21    else {
22        fairfinal = fair; // no discount
23        std::cout << "Regular Fair: " << fairfinal << " Pesos" << std::endl;
24    }
25
26    return 0;
27 }
```
- Toolbar:** Includes icons for copy, paste, share, and run.
- Run Button:** A blue "Run" button is visible in the toolbar.
- Output Area:**
 - Shows the command: Enter Your Age: 69
 - Shows the result: Senior Discount Fair: 8.1 Pesos
 - Shows the message: === Code Execution Successful ===

The screenshot shows a dark-themed online C++ compiler interface. On the left, the code editor contains a main.cpp file with the following content:

```
1 #include <iostream>
2
3 int main() {
4     int age;
5     double fair = 9.0;
6     double fairfinal;
7
8     std::cout << "Enter Your Age: ";
9     std::cin >> age;
10
11    if (age >= 60) { // 60 years old is considered being a senior citizen
12        // (Philippines).
13        fairfinal = fair - (fair * 0.10);
14        std::cout << "Senior Discount Fair: " << fairfinal << " Pesos" << std::endl;
15    } else if (age <= 25) { // anyone could be still studying at a later age but I'm
16        // going with this
17        fairfinal = fair - (fair * 0.08);
18        std::cout << "Student Discount Fair: " << fairfinal << " Pesos" << std::endl;
19    } else {
20        fairfinal = fair; // no discount
21        std::cout << "Regular Fair: " << fairfinal << " Pesos" << std::endl;
22    }
23
24
25    return 0;
26 }
```

At the top right, there are icons for copy, share, and run, followed by a 'Run' button and a 'Clear' button. The output window on the right displays the results of running the code with an input of 12:

```
Enter Your Age: 12
Student Discount Fair: 8.28 Pesos

*** Code Execution Successful ***
```

This screenshot shows a similar dark-themed online C++ compiler interface. The code editor contains the same main.cpp file as the first one.

```
1 #include <iostream>
2
3 int main() {
4     int age;
5     double fare = 9.0;
6     double fareFinal;
7
8     std::cout << "Enter Your Age: ";
9     std::cin >> age;
10
11    if (age >= 60) { // 60 years old is consider being a senior citizen
12        // (Philippines).
13        fareFinal = fare * 0.10;
14        std::cout << "Senior Discount Fare: " << fareFinal << " Pesos"
15        << std::endl;
16    } else if (age <= 25) { // any one could be still studying at a later
17        // age but im going with this
18        fareFinal = fare * 0.50;
19    }
20
21    return 0;
22 }
```

The output window shows the results of running the code with an input of 12:

```
Enter Your Age: 12
Student Discount Fare: 4.5 Pesos

*** Code Execution Successful ***
```

```

main.cpp
1 #include <iostream>
2
3 int main() {
4     int age;
5     double fair = 9.0;
6     double fairfinal;
7
8     std::cout << "Enter Your Age: ";
9     std::cin >> age;
10
11    if (age >= 60) { // 60 years old is considered being a senior citizen (Philippines).
12        fairfinal = fair - (fair * 0.10);
13        std::cout << "Senior Discount Fair: " << fairfinal << " Pesos" << std::endl;
14    }
15    else if (age <= 25) { // anyone could be still studying at a later age but I'm going with
16        this
17        fairfinal = fair - (fair * 0.08);
18        std::cout << "Student Discount Fair: " << fairfinal << " Pesos" << std::endl;
19    }
20    else {
21        fairfinal = fair; // no discount
22        std::cout << "Regular Fair: " << fairfinal << " Pesos" << std::endl;
23    }
24
25    return 0;
26 }

```

Output

```

Enter Your Age: 30
Regular Fair: 9 Pesos

--- 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.

Initialize total to zero

Initialize counter to zero

Input the first grade

While the user has not 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"

CODE:

```
#include int main() {
int grade;
```

```

int total = 0;
int counter = 0;
double average;

std::cout << "Enter a grade (-1 to end): ";
std::cin >> grade;
while (grade != -1) {
    total = total + grade;
    counter = counter + 1;

    std::cout << "Enter a grade (-1 to end): ";
    std::cin >> grade;
}

if (counter != 0) {
    average = static_cast(total) / counter;
    std::cout << "Class average is " << average << std::endl;
} else {
    std::cout << "No grades were entered." << std::endl;
}

}

Return 0;
}

```

The screenshot shows a code editor window with the following details:

- Title Bar:** The title bar displays "main.cpp".
- Toolbar:** A toolbar with icons for copy, paste, share, and run.
- Code Area:** The code area contains the C++ program provided in the question.
- Output Area:** The output area shows the execution results:


```

Enter a grade (-1 to end): 99
Enter a grade (-1 to end): 88
Enter a grade (-1 to end): 96
Enter a grade (-1 to end): 88
Enter a grade (-1 to end): 98
Enter a grade (-1 to end): -1
Class average is 93.8

==== Code Execution Successful ====

```

8. Conclusion:

Understand and use conditions and conditional executions with the *if/else* selection structure. Understand and use conditions and conditional executions with the *while* selection structure

9. Assessment Rubric

Rubric for SO 7 (7)							
Criteria				Ratings			Pts
④ SO 7 PI 1 ILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Educational interests and pursuits exist and flourish outside classroom requirements, knowledge and/or experiences are pursued independently and applies knowledge learned into practice	5 pts Good Educational interests and pursuits exist and flourish outside classroom requirements, knowledge and/or experiences are pursued independently	4 pts Satisfactory Look beyond classroom requirements, showing interest in pursuing knowledge independently	3 pts Unsatisfactory Begins to look beyond classroom requirements, showing interest in pursuing knowledge independently	2 pts Poor Relies on classroom instruction only	1 pts Very Poor No initiative or interest in acquiring new knowledge	6 pts
④ SO 7 PI 2 ILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Completes an assigned task independently and practices continuous improvement	5 pts Good Completes an assigned task without supervision or guidance	4 pts Satisfactory Requires minimal guidance to complete an assigned task	3 pts Unsatisfactory Requires detailed or step-by-step instructions to complete a task	2 pts Poor Shows little interest to complete a task independently	1 pts Very Poor No interest to complete a task independently	6 pts
④ SO 7 PI 3 ILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Synthesizes and integrates information from a variety of sources; formulates a clear and precise perspective; draws appropriate conclusions	5 pts Good Evaluate information from a variety of sources; formulates a clear and precise perspective.	4 pts Satisfactory Analyze information from a variety of sources; formulates a clear and precise perspective.	3 pts Unsatisfactory Apply the gathered information to formulate the problem	2 pts Poor Gather and summarized the information from a variety of sources but failed to formulate the problem	1 pts Very Poor Gather information from a variety of sources	6 pts
④ SO 7 PI 4 ILO4 Utilize lifelong learning skills in pursuit of personal development and excellence in professional practice. threshold: 4.8 pts	6 pts Excellent Ideas are combined in original and creative ways in line with the new and emerging technology trends to solve a problem or address an issue.	5 pts Good Ideas are creative and adapt the new knowledge to solve a problem or address an issue	4 pts Satisfactory Ideas are creative in solving a problem, or address an issue	3 pts Unsatisfactory Shows some creative ways to solve the problem	2 pts Poor Shows initiative and attempt to develop creative ideas to solve the problem	1 pts Very Poor Ideas are copied or restated from the sources consulted	6 pts

Total Points: 24