

Activity No. 2.2	
Control Structures (part 1)	
Course Code: CPE 007	Program: Computer Engineering
Course Title: Programming Logic and Design	Date Performed: 8/11/25
Section: CPE11S1	Date Submitted: 8/11/25
Name(s): Paula G Esguerra	Instructor: Engr. Jimlord Quejado
6. Output	
<p>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.</p> <p>Using the following pseudocode the program can be as follows:</p> <p>Set total to zero</p> <p>Set grade counter to one</p> <p>While grade counter is less than or equal to ten</p> <p> Input the next grade</p> <p> Add the grade into the total</p> <p> Add one to the grade counter</p> <p>Set the class average to the total divided by ten</p> <p>Print the class average</p>	
<p>Counter controlled repetition is often called definite repetition because the number of repetitions is known before the loop begins executing. In the example above, the number of grades (10) was known in advance. What if there is no indication is given of how many grades are to be entered? How can the program determine when to stop the input of grades? How it will know when to calculate and print the class average?</p> <p>One way to solve this is to use a special value called the sentinel value (also called a signal value, a dummy xvalue or a flag value) to indicate “end of data entry”. The user types grades in until all legitimate grades have been entered. The user types the sentinel value to indicate that the last grade has been entered. Sentinel-controlled repetition is often called indefinite repetition because the number of repetitions is not known before the loop begins executing.</p>	

main.cpp		Run	Output
1 <code>#include <iostream></code>			Enter grade 1: 2
2			Enter grade 2: 3
3 <code>int main() {</code>			Enter grade 3: 45
4 <code> int total = 0;</code>			Enter grade 4: 26
5 <code> int grade;</code>			Enter grade 5: 30
6 <code> int grade_counter = 1;</code>			Enter grade 6: 46
7			Enter grade 7: 47
8 <code> while (grade_counter <= 10) {</code>			Enter grade 8: 30
9 <code> std::cout << "Enter grade " << grade_counter <<</code>			Enter grade 9: 70
10 <code> ":";</code>			Enter grade 10: 80
11 <code> std::cin >> grade;</code>			Class average is: 37.9
12 <code> // Optional: Validate input range</code>			
13 <code> if (grade < 0 grade > 100) {</code>			==== Code Execution Successful ===
14 <code> std::cout << "Invalid grade. Please enter a</code>			
15 <code> value between 0 and 100.\n";</code>			
16 <code> continue; // skip incrementing counter if</code>			
17 <code> invalid input</code>			
18 <code> }</code>			
19 <code> total += grade;</code>			
20 <code> grade_counter++;</code>			
21 <code>}</code>			
22 <code>double average = static_cast<double>(total) / 10;</code>			
23			
24 <code>std::cout << "Class average is: " << average << std</code>			
25 <code> ::endl;</code>			
26 <code>return 0;</code>			
27 <code>}</code>			
28			
29			

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.

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"

/*Class average program with counter-controlled repetition */

```
#include<iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    int counter, grade, total, average;
```

```
    /* initialization phase */
```

```
    total = 0;
```

```
    counter = 1;
```

```
    /* processing phase */
```

```
    while (counter <=10){
```

```
        cout<<"Enter the grade: ";
```

```
        cin>>grade;
```

```
        total = total + grade;
```

```
        counter = counter + 1;
```

```
}
```

```
    /* termination phase */
```

```

average = total / 10;

cout<<"Class average is" <<average;

return 0; /* program ends */

}

```

main.cpp

```

1 #include <iostream>
2
3 int main() {
4     int total = 0;
5     int grade;
6     int counter = 0;
7
8     // Input the first grade
9     std::cout << "Enter a grade between 0 and 100 (or -1 to finish):";
10    std::cin >> grade;
11
12    // While the user has not as yet entered the sentinel
13    while (grade != -1) {
14        // Check if even or odd
15        if (grade % 2 == 0) {
16            std::cout << grade << " is an even number. " << std::endl;
17        } else {
18            std::cout << grade << " is an odd number. " << std::endl;
19        }
20
21        // Add to total and increment counter
22        total += grade;
23        counter++;
24
25        // Input the next grade
26        std::cout << "Enter a grade between 0 and 100 (or -1 to finish): ";
27        std::cin >> grade;
28    }
29
30    // After loop ends, check if any grades were entered
31    if (counter != 0) {
32        double average = static_cast<double>(total) / counter;
33        std::cout << "Average of grades: " << average << std::endl;
34    } else {
35        std::cout << "No grades were entered. " << std::endl;
36    }
37
38
39    return 0;
40 }
41

```

Output

```
Enter a grade between 0 and 100 (or -1 to finish):70
70 is an even number.
Enter a grade between 0 and 100 (or -1 to finish): 80
80 is an even number.
Enter a grade between 0 and 100 (or -1 to finish): 90
90 is an even number.
Enter a grade between 0 and 100 (or -1 to finish): -1
Average of grades: 80
```

```
==== 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.

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"

```
/*Class average program with counter-controlled repetition */

#include<iostream>

using namespace std;

int main()

{

    int counter, grade, total, average;

    /* initialization phase */

    total = 0;

    counter = 1;

    /* processing phase */

    while (counter <=10){

        cout<<"Enter the grade: ";

        cin>>grade;

        total = total + grade;

        counter = counter + 1;

    }

    /* termination phase */

    average = total / 10;

    cout<<"Class average is" <<average;

    return 0; /* program ends */

}
```

main.cpp



```
1 #include <iostream>
2
3
4 int main() {
5     const double MIN_FARE = 9.0;
6     int age;
7     double discount = 0.0;
8     double finalFare;
9
10    // Ask user for age
11    std::cout << "Enter your age: ";
12    std::cin >> age;
13
14    // Apply discount using conditional statements
15    if (age >= 60) {
16        discount = 0.10; // 10% discount for seniors
17    } else if (age >= 5 && age <= 21) {
18        discount = 0.08; // 8% discount for students
19    } else {
20        discount = 0.0; // No discount
21    }
22
23    // Calculate final fare
24    finalFare = MIN_FARE - (MIN_FARE * discount);
25
26    // Display results
27    std::cout << "Discount applied: " << (discount * 100) << "%" << std::endl;
28    std::cout << "Fare to pay: " << finalFare << " pesos" << std::endl;
29
30    return 0;
31 }
32
```

Output

```
Enter your age: 70
Discount applied: 10%
Fare to pay: 8.1 pesos
```

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

Output

```
Enter your age: 17
Discount applied: 8%
Fare to pay: 8.28 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.

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"

```
/*Class average program with counter-controlled repetition */  
  
#include<iostream>  
  
using namespace std;  
  
int main()  
{  
    int counter, grade, total, average;  
  
    /* initialization phase */  
  
    total = 0;  
  
    counter = 1;  
  
    /* processing phase */  
  
    while (counter <=10){
```

```
cout<<"Enter the grade: ";
cin>>grade;
total = total + grade;
counter = counter + 1;
}
/* termination phase */
average = total / 10;
cout<<"Class average is" <<average;
return 0; /* program ends */
}
```

main.cpp



Share

Run

```
1 #include <iostream>
2
3 int main() {
4     int total = 0;
5     int counter = 0;
6     int grade;
7
8     // Input the first grade
9     std::cout << "Enter grade (-1 to finish): ";
10    std::cin >> grade;
11
12    // While the user has not entered the sentinel
13    while (grade != -1) {
14        total += grade;           // Add to total
15        counter++;              // Increment counter
16
17        // Input the next grade (possibly the sentinel)
18        std::cout << "Enter grade (-1 to finish): ";
19        std::cin >> grade;
20    }
21
22    // If at least one grade was entered
23    if (counter != 0) {
24        double average = static_cast<double>(total) / counter;
25        std::cout << "Class average is: " << average << std::endl;
26    } else {
27        std::cout << "No grades were entered." << std::endl;
28    }
29
30    return 0; // program ends
31 }
```

Output

```
Enter grade (-1 to finish): 2
Enter grade (-1 to finish): 5
Enter grade (-1 to finish): 6
Enter grade (-1 to finish): 7
Enter grade (-1 to finish): 8
Enter grade (-1 to finish): 4
Enter grade (-1 to finish): 2
Enter grade (-1 to finish): -1
Class average is: 4.85714
```

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

8. Conclusion

The if/else is used for evaluation of a condition, for example, == (equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to), and etc... This if/else maybe always be basing on the condition's truth value. The while loop executes a specified condition that remains true, it uses the same conditions if/else use. The counter-controlled repetition is when the number of iterations is known in advance, tested against a condition and being updated. Sentinel-controlled repetition is the somehow the opposite of counter, is used when the number of iterations is unknown and depends on a specific condition and input. I learned how to use and mixing them into one code, if/else, while loop, and the repetitions.

9. Assessment Rubric

Rubric for SO 7 (7)							
Criteria	Ratings						Pts
◎ SO 7 PI 1 IILO4 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 pt
◎ SO 7 PI 2 IILO4 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 pt
◎ SO 7 PI 3 IILO4 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 pt
◎ SO 7 PI 4 IILO4 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 pt

Total Points: 24