Sadiq Public School

Subject: Computer **Class: I2** **Saturday, 16th November 2024**

Lesson : Do-While loop Implementation and working

A: Inquiry

What is a do-while Loop?

A do-while loop is a type of control flow structure in programming that allows a set of instructions to be executed repeatedly based on a condition. Unlike other loops (for and while), the do-while loop **executes the code block at least once**, regardless of the condition, because the condition is checked **after** the code block runs. This guarantees that the loop's body will execute a minimum of one time.

The syntax of a do-while loop in C is as follows:

```
do {
   // code to be executed
} while (condition);
```

B: Information

Why Use a do-while Loop?

There are specific scenarios where a do-while loop is advantageous:

1. **Guaranteed Execution:** If you need a block of code to run at least once, regardless of whether the condition is true initially, a do-while loop is ideal. This is useful when the first execution of the code block might set up conditions for subsequent executions.
2. **User Input Validation:** It's commonly used in situations where you need to prompt the user for input, and you want to repeat the prompt until the input is valid. The loop ensures the user is prompted at least once before checking if the input meets specific criteria.
3. **Menu-Driven Programs:** In interactive programs that use menus (e.g., displaying options to users and asking for their choice), a do-while loop allows displaying the menu and processing the user's choice until they decide to exit.

How Does a do-while Loop Work?

1. **Initial Execution:** The loop starts by executing the code block inside the do section. This is different from a while or for loop, where the condition is checked before the first execution.
2. **Condition Check:** After running the code block, the loop checks the condition in the while part. If the condition is **true**, the loop will repeat, executing the code block again.
3. **Termination:** If the condition becomes **false**, the loop stops, and the program continues with the next line of code after the do-while loop.

Key Differences between do-while and while Loops

Feature	do-while Loop	while Loop
Condition Check	After executing the loop body	Before executing the loop body
Guaranteed Execution	Yes, executes at least once	No, might not execute if condition is false initially
Common Usage	Input validation, menu-driven programs	Repeating tasks with pre-known conditions

When to Use a do-while Loop?

Use a do-while loop when you need the code to execute at least once, or when you want to check the condition after running the code. It's especially useful in scenarios involving user interaction, where you may want to prompt users at least once before validating their input.

Sample Programs Using the do-while Loop:

1. **Program to Print Even Numbers within a Range Entered by the User**

```
#include <stdio.h>

int main() {
  int start, end;
  printf("Enter the starting number: ");
  scanf("%d", &start);
  printf("Enter the ending number: ");
  scanf("%d", &end);

  int num = start;
  do {
    if (num % 2 == 0) {
            printf("%d ", num);
        }
        num++;
    } while (num <= end);

    return 0;
}
```

Explanation:

- The program first prompts the user to enter a starting and an ending number, which are stored in the variables start and end.
- The variable num is initialized to the starting number (start).
- Inside the do-while loop, the program checks if num is even by using the condition num % 2 == 0. If the condition is true, num is printed as it's an even number.
- The num variable is then incremented by 1, moving to the next number in the range.
- The loop continues until num exceeds the end value.
- This ensures that only even numbers between the starting and ending values are printed.

2. **Program to Print the Multiplication Table of a Given Number**

```
#include <stdio.h>

int main() {
    int num, i = 1;
    printf("Enter a number to print its multiplication table: ");
    scanf("%d", &num);

    do {
        printf("%d x %d = %d\n", num, i, num * i);
        i++;
    } while (i <= 10);

    return 0;
}
```

Explanation:

- The user inputs a number (num) for which the multiplication table will be printed.
- The variable i is initialized to 1, representing the multiplier in the table.
- In each iteration of the do-while loop, the program calculates and displays the result of num * i.
- After printing each result, i is incremented by 1.
- The loop continues until i becomes greater than 10, meaning the table has been printed up to num x 10.

3. **Program to Display Series 1 + 1/2 + 1/4 + 1/6 + ... + 1/100**

```
#include <stdio.h>

int main() {
   double sum = 0.0;
   int i = 1;

   do {
      sum += 1.0 / i;
      i += 2; // increments by 2 to get 1, 1/2, 1/4, etc.
   } while (i <= 100);

   printf("The sum of the series is: %lf\n", sum);

   return 0;
}
```

Explanation:

- This program calculates the sum of a series where the denominator increases by 2 in each term.
- The sum variable (of type double to store decimal values) is initialized to 0.0.
- The variable i starts at 1, representing the denominator of the first term.
- Inside the do-while loop, each term 1.0 / i is added to sum.
- i is then incremented by 2 to get the next denominator in the series (1, 2, 4, 6, …, up to 100).
- The loop ends once i exceeds 100, and the final sum is printed.

4. **Program to Display Series 3, 5, 7, 9, 11, 13, ... , 23**

```
#include <stdio.h>

int main() {
   int num = 3;

   do {
      printf("%d ", num);
      num += 2;
   } while (num <= 23);

   return 0;
}
```

Explanation:

- The program starts by initializing num to 3, the first number in the series.
- Inside the do-while loop, the current value of num is printed.

- num is then incremented by 2 to get the next odd number in the series (5, 7, 9, etc.).
- The loop continues until num exceeds 23.
- This sequence ensures the series 3, 5, 7, ..., 23 is printed in its entirety.

Each program demonstrates the use of the do-while loop to solve specific tasks. This loop structure ensures that the code within the loop executes at least once, which is beneficial in cases where we need at least one operation to occur regardless of the initial condition. Each program's logic is carefully crafted to match its task, utilizing the do-while loop's unique structure effectively.

Topics to cover:
Book: Computer Science 12 MS Access and C by IT Series
Chapter # 12 : loop constructs

- **Additional Resources**: Watch this video for an introduction to the while loop in programming

  Working of Do-while loop

C: Synthesizing/absorbing the information

Read the lesson in your notes or textbook, and answer the following questions in your notebook:

i. **What is the purpose of a while loop?** Explain how it differs from other types of loops.

ii. **Explain the syntax of a while loop**. Why is it important for the condition to be carefully controlled?

iii. **Write down the steps** for the example of printing numbers from 1 to 5. Show how the value of i changes with each iteration.

D: Practicing

- **Write a program using a do-while loop** to print all odd numbers from 1 to 50.
- **Modify the table program** to print the multiplication table up to 20 instead of 10.
- **Write a program to calculate the factorial of a given number** using a do-while loop.
- **Explain the difference between the while and do-while loops** in a short paragraph and give an example of when each might be used

Note: All the students have to complete their home assignments from Wednesday to Saturday in their notebooks

Feedback:
Students: Please if you have any questions at all about this topic, any words you don't understand, anything at all, please send your concerned teacher an email and you will get a reply a.s.a.p.

Class	Teacher Name	Teacher Abbreviation	Email Address	Instruction
I2C	Bilal Mustafa Khan	BMK	Bilal.rohaila@gmail.com	Students of I2C will correspond with BMK in case of any query.
I2D	Moazzam Iqbal	MI	Moazzam_iqball@hotmail.com	Students of I2D will correspond with MI in case of any query.
I2GB	Sara Naheed	SN	Saranaheed.9201@gmail.com	Students of I2GB will correspond with SN in case of any query.