CC-112L

Programming Fundamentals

Laboratory 02

Introduction to Programming, Algorithms and C

Version: 1.0.0

Release Date: 18-02-2025

Department of Information Technology

University of the Punjab

Lahore, Pakistan

Contents:

Learning Objectives:

- Understand and implement different **control structures** in C.
- Learn how **nested control structures** work and their applications.
- Write C programs using **decision-making** and **looping constructs**.

Resources Required:

- Desktop Computer or Laptop
- Microsoft ® Visual Studio 2022

General Instructions:

- In this Lab, you are **NOT** allowed to discuss your solution with your colleagues, even not allowed to ask how is s/he doing, this may result in negative marking. You can **ONLY** discuss with your Teaching Assistants (TAs) or Lab Instructor.
- Your TAs will be available in the Lab for your help. Alternatively, you can send your queries via email to one of the followings.

Teachers:		
Course Instructor	Hafiz Anzar Ahmad	anzar@pucit.edu.pk
Teacher Assistants	Manahil	Bitf21m002@pucit.edu.pk

Overview

1. Control Structures in C

Control structures determine the flow of execution in a program. There are three main types:

1. **Sequential Control** – Executes statements in the order they appear.
2. **Selection (Decision-Making) Control** – Executes specific code based on conditions.
3. **Iteration (Looping) Control** – Repeats code blocks based on conditions.

---

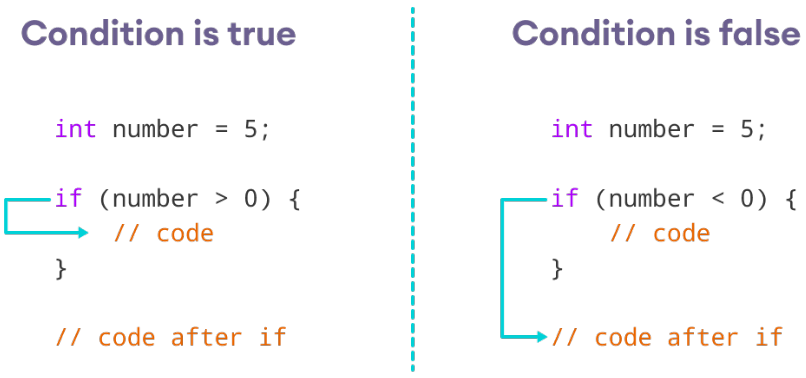
2. Selection (Decision-Making) Control Structures

These structures allow the program to make decisions based on conditions.

(i) if Statement

Executes a block of code only if a condition is true.

```
if (condition) {
   // Code to execute if condition is true
}
```



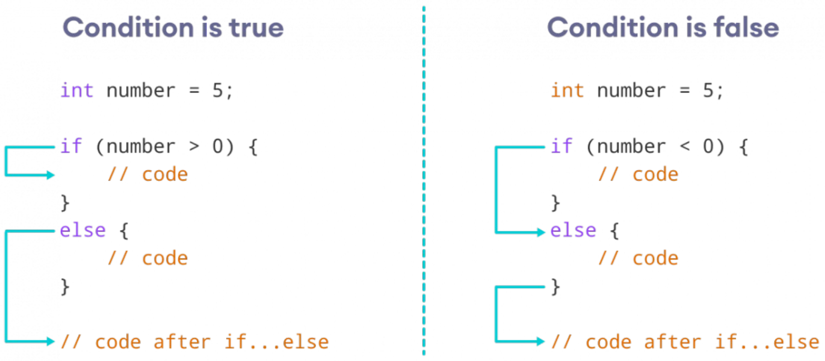
```
int number = 5;

if (number > 0) {
    // code
}

// code after if
```

```
int number = 5;

if (number < 0) {
    // code
}

// code after if
```

(ii) if-else Statement

Executes one block if the condition is true, another if it's false.

```
if (condition) {
   // Code if condition is true
} else {
   // Code if condition is false
}
```



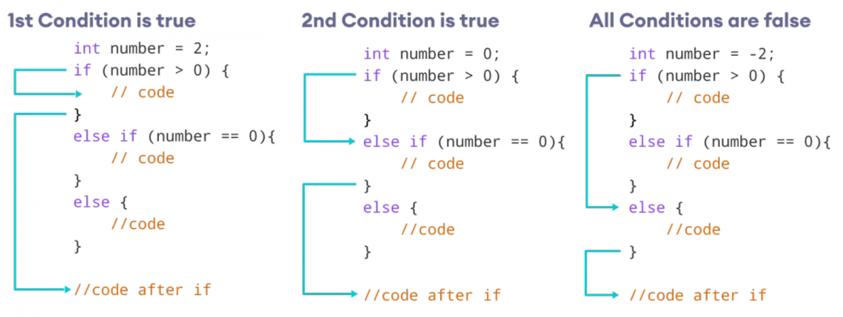
```
int number = 5;

if (number > 0) {
    // code
}
else {
    // code
}

// code after if...else
```

```
int number = 5;

if (number < 0) {
    // code
}
else {
    // code
}

// code after if...else
```

(iii) else-if Ladder

Checks multiple conditions in sequence.

```
if (condition) {
    // Code if condition is true
} else if (condition) {
    // Code if condition is false
} else {
    // Code if condition is false
}
```



```
int number = 2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

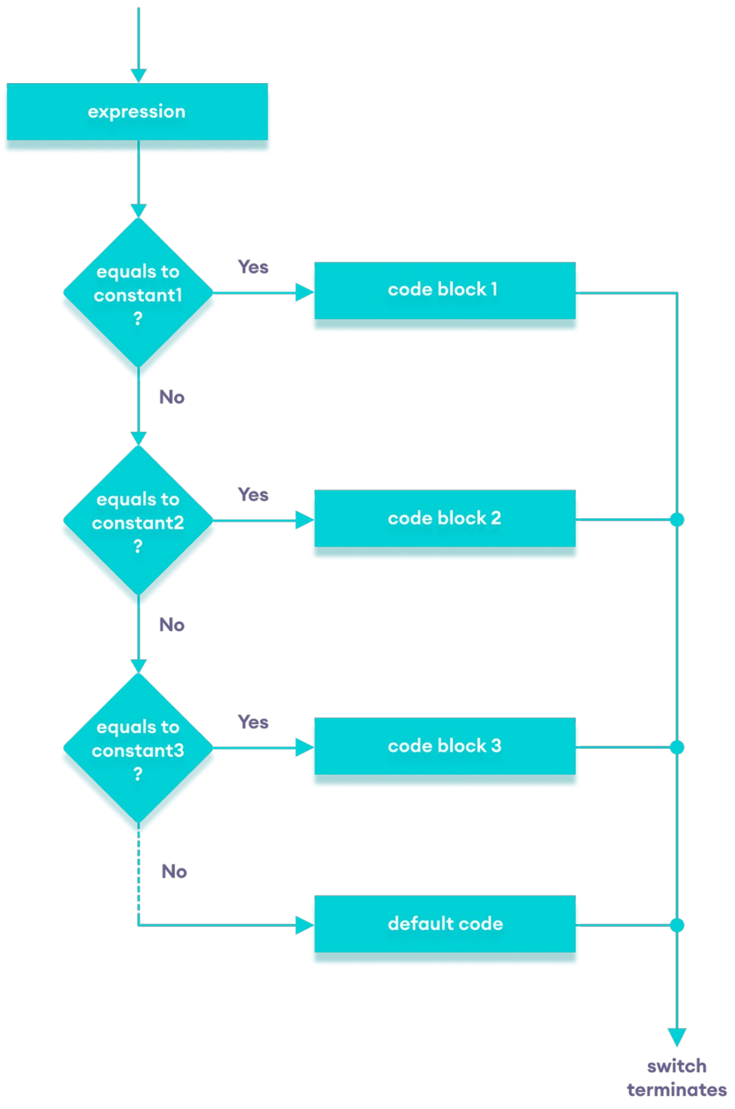
```
int number = 0;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

```
int number = -2;
if (number > 0) {
    // code
}
else if (number == 0){
    // code
}
else {
    //code
}

//code after if
```

(iv) switch Statement

Used when multiple values of a variable need to be checked.

```
switch (variable) {
    case value1:
        // Code for case 1
        break;
    case value2:
        // Code for case 2
        break;
    default:
        // Code if no case matches
}
```



3. Iteration (Looping) Control Structures

Loops execute a block of code multiple times.

(i) for Loop

Executes a loop for a fixed number of iterations.

```
for (initialization; condition; increment/decrement) {
    // Code inside loop
}
```

Example:

```
for (int i = 1; i <= 5; i++) {
    printf("%d ", i);
}
```

output:

```
1 2 3 4 5
```

(ii) while Loop

Executes a loop while a condition remains true.

```
while (condition) {
    // Code inside loop
}
```

Example:

```
int i = 1;
while (i <= 5) {
    printf("%d ", i);
    i++;
}
```

output:

```
1 2 3 4 5
```

(iii) do-while Loop

Executes at least once before checking the condition.

```
do {
     // Code inside loop
} while (condition);
```

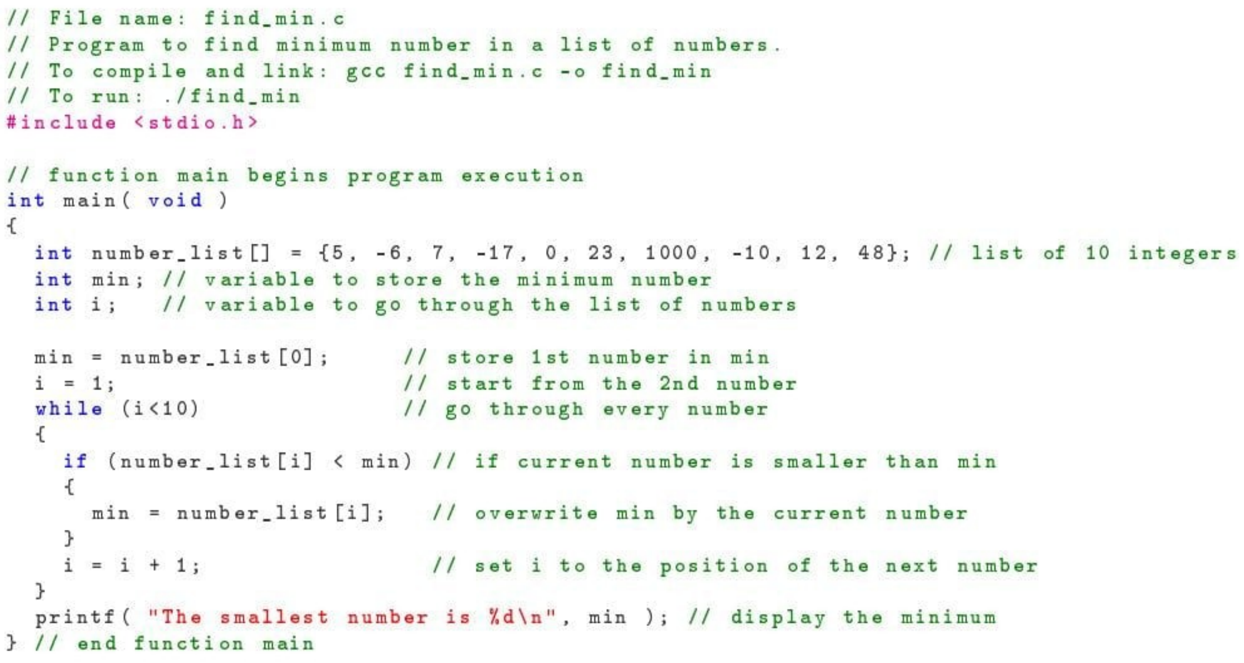
Example:

```
int i = 1;
do {
    printf("%d ", i);
    i++;
} while (i <= 5);
```

output:

```
1 2 3 4 5
```

C program to find the minimum of a list of numbers



```
// File name: find_min.c
// Program to find minimum number in a list of numbers.
// To compile and link: gcc find_min.c -o find_min
// To run: ./find_min
#include <stdio.h>

// function main begins program execution
int main( void )
{
  int number_list[] = {5, -6, 7, -17, 0, 23, 1000, -10, 12, 48}; // list of 10 integers
  int min; // variable to store the minimum number
  int i;   // variable to go through the list of numbers

  min = number_list[0];     // store 1st number in min
  i = 1;                    // start from the 2nd number
  while (i<10)              // go through every number
  {
    if (number_list[i] < min) // if current number is smaller than min
    {
      min = number_list[i];   // overwrite min by the current number
    }
    i = i + 1;                // set i to the position of the next number
  }
  printf( "The smallest number is %d\n", min ); // display the minimum
} // end function main
```

---

4. Nested Control Structures

A control structure inside another control structure is called **nesting**.

(i) Nested if Statements

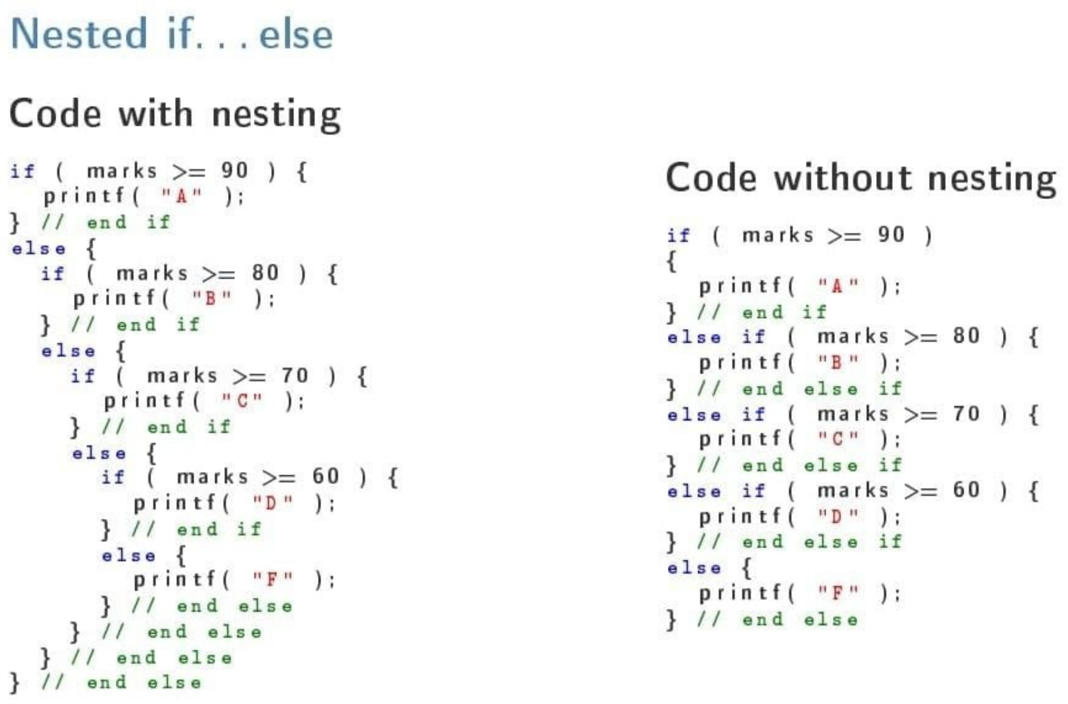
```
if (condition1) {
   if (condition2) {
      // Code executes if both conditions are true
   }
}
```

(i) Nested if-else Statements

```
if (condition1) {
   if (condition2) {
      // Code executes if both condition1 and condition2 are true
   } else {
      // Code executes if condition1 is true but condition2 is false
   }
} else {
   // Code executes if condition1 is false
}
```

(iii) Nested switch Statements

```
switch (var1) {
   case value1:
      switch (var2) {
         case value2:
            // Nested switch case
            break;
      }
      break;
}
```



Nested if. . . else

Code with nesting

```
if ( marks >= 90 ) {
   printf( "A" );
} // end if
else {
   if ( marks >= 80 ) {
      printf( "B" );
   } // end if
   else {
      if ( marks >= 70 ) {
         printf( "C" );
      } // end if
      else {
         if ( marks >= 60 ) {
            printf( "D" );
         } // end if
         else {
            printf( "F" );
         } // end else
      } // end else
   } // end else
} // end else
```

Code without nesting

```
if ( marks >= 90 )
{
   printf( "A" );
} // end if
else if ( marks >= 80 ) {
   printf( "B" );
} // end else if
else if ( marks >= 70 ) {
   printf( "C" );
} // end else if
else if ( marks >= 60 ) {
   printf( "D" );
} // end else if
else {
   printf( "F" );
} // end else
```

whats the difference…?

1. Code with Nesting (Left Side)

- The conditions are **checked sequentially inside separate `else` blocks**.
- If a condition is false, control goes **inside another if-else block**.
- This increases **indentation depth**, making the code harder to read.
- **Example:**
    - If `marks = 85`, the first condition (`marks >= 90`) fails.
    - It enters the `else` block, where it checks `marks >= 80`, which is true.
    - So, `"B"` is printed.

2. Code without Nesting (Right Side)

- Uses an **else-if ladder**, meaning conditions are checked **in a single block**.
- The program checks conditions **one by one in a single structure**, making it easier to read.
- **Example:**
    - If `marks = 85`, it fails `marks >= 90`, moves to `marks >= 80`, which is true.
    - `"B"` is printed.
    - No further conditions are checked.

PRE-LAB TASKS

Task 1

Write a C program to compute the sum of the first 10 odd natural numbers.

Expected Output:

```
12345678910
The Sum is : 55
```

Task 2

Write a C program to print the week day by taking an integer from user in range 1-7 else print invalid day.

Test Data :
N : 3
Expected Output :
day: wednesday

Task 3

Write a C program to convert specified days into years, weeks and days.

Test Data :
Number of days : 1329
Expected Output :

```
Years: 3
Weeks: 33
Days: 3
```

Task 4

Write a C program to print the table of given number n.

Test data :
Input n =2
Expected Output :

```
2 × 1 = 2
...............
2 ×10 = 20
```

Task 5

Write a C program to read the roll no, name and marks of three subjects and calculate the total, percentage and division.

Test Data

```
Input the Roll Number of the student :784
Input the Name of the Student :James
Input the marks of Physics, Chemistry and
Computer Application : 70 80 90
```

Expected Output

```
Roll No :784
Name of Student: James
Marks in Physics : 70
Marks in Chemistry : 80
Marks in Computer Application : 90
Total Marks = 240
Percentage = 85.00
Division: first
```

Division :

First	>=85
Second	>=65
Third	>=45
Else, Fail	

Submissions:

- For Pre-Lab Activity: Submit the .c file on Google Classroom and name it to your roll no.

References and Additional Material: