كليـــة العلـــــوم  
قسم الأمن السيبراني

**Subject: Programming Fundamentals**

**First Stage**

**Lecturer: Dr. Abdulkadhem A. Abdulkadhem**

**Lecture (5)**

**Selection Statements**

**Lecture Outline**

1. **Introduction to Selection Statements**
2. **Types of Selection Statements**
   * Single if Statement
   * Single if Block Statement
   * Single if-else Statement
   * else if Statement
   * Nested if Statements
3. **Examples and Explanations**

**1. Introduction to Selection Statements**

**Definition:** Selection statements allow a program to make decisions and execute specific blocks of code based on conditions.

**Importance:**

* Enables conditional logic in programs.
* Allows decision-making to adapt to input or runtime data.
* Provides flexibility and control over program execution.

**2. Types of Selection Statements**

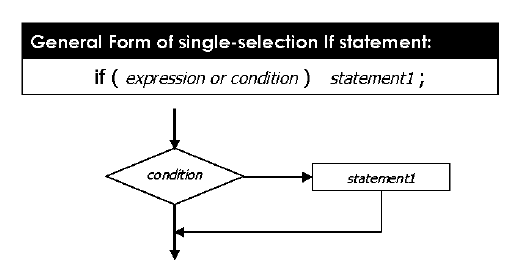
**A. Single if Statement Structure**

**Syntax:**

|  |
| --- |
| if (condition) {  // Code to execute if the condition is true  } |

**Explanation:**

* The condition is evaluated. If it is true (non-zero), the code inside the if block executes.
* If false (zero), the block is skipped.



**Example 1:**

|  |
| --- |
| int x = 10;  if (x > 5)  cout << "x is greater than 5"; |

**Explanation:**

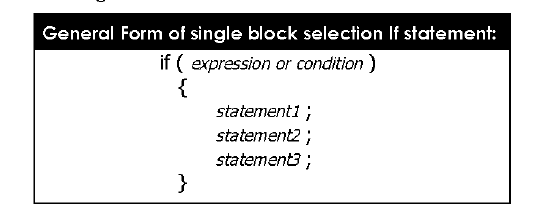
* The condition x > 5 evaluates to true, so the message is printed.

**Example 2:**

|  |
| --- |
| int temperature = -5;  if (temperature < 0)  cout << "It's freezing!"; |

**Explanation:**

* The condition temperature < 0 evaluates to true, so the message is printed.

**B. Single if Block Statement Structure**

**Example 1:**

|  |
| --- |
| int score = 85;  if (score >= 50) {  cout << "You passed!\n";  cout << "Congratulations!";  } |

**Explanation:**

* Both lines inside the block execute because score >= 50 is true.

**Example 2:**

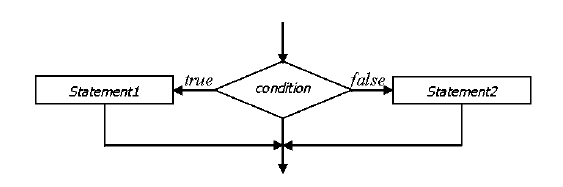
|  |
| --- |
| int age = 18;  if (age >= 18) {  cout << "You are an adult.\n";  cout << "You can vote.";  } |

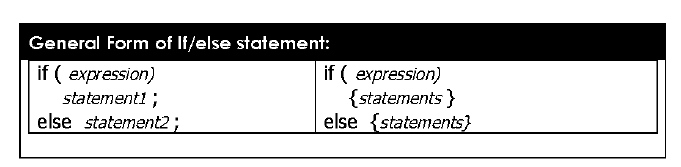
**Explanation:**

* The condition age >= 18 evaluates to true, so both messages are displayed.

**C. Single if-else Statement Structure**

**Syntax:**





**Example 1:**

|  |
| --- |
| int number = 10;  if (number % 2 == 0) {  cout << "Even number";  } else {  cout << "Odd number";} |

**Explanation:**

* If number is divisible by 2, it prints "Even number"; otherwise, "Odd number".

**Example 2:**

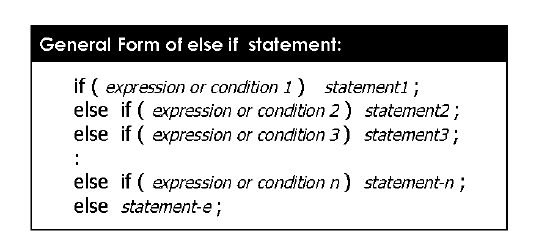
|  |
| --- |
| int marks = 45;  if (marks >= 50) {  cout << "You passed!";  } else {  cout << "You failed.";  } |

**Explanation:**

* The marks >= 50 condition determines which message is displayed.

**D. else if Statement Structure**

**Syntax:**



**Example 1:**

|  |
| --- |
| int grade = 85;  if (grade >= 90) {  cout << "Excellent";  } else if (grade >= 75) {  cout << "Good";  } else {  cout << "Needs Improvement";  } |

**Explanation:**

* The program checks each condition in sequence and executes the block for the first true condition.

**Example 2:**

|  |
| --- |
| int temperature = 15;  if (temperature > 30) {  cout << "Hot";  } else if (temperature > 20) {  cout << "Warm";  } else {  cout << "Cold";  } |

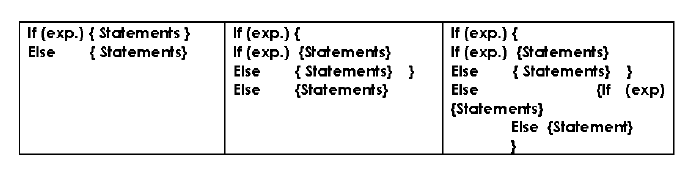
**Explanation:**

* Based on temperature, the appropriate message is displayed.

**E. Nested if Statement Structure**

Some of the samples of *NESTED if-else* constructions are shown below:

**Syntax:**

****

**Example 1:**

|  |
| --- |
| int x = 10, y = 20;  if (x > 5) {  if (y > 15) {  cout << "x is greater than 5 and y is greater than 15";  }  } |

**Explanation:**

* The outer if checks x > 5, and the inner if checks y > 15.

**Example 2:**

|  |
| --- |
| int age = 25;  string gender = "female";  if (age > 18) {  if (gender == "female") {  cout << "You are an adult female.";  }  } |

**Explanation:**

* The nested if checks both age > 18 and gender == "female" before printing the message.