	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

Experiment: 3

PART A

(PART A: TO BE REFERRED BY STUDENTS)

Aim: Implementing programs using conditional/Decision making/selection statements

Learning Outcomes: The learner would be able to

1. Identify appropriate decision-making statement
2. Understand the syntax of decision-making statements
3. Use decision-making statements to solve problems by writing programs
4. Work with nested decision-making statements

Theory:

C++ Control Constructs/Structure/Statements

- Control statements are used to alter the flow of program execution.
- Control statements evaluate the condition (uses relational and/or logical operators) & control the flow of execution.
- C++ control constructs/statements are as follows.

Decision Making Statements or Conditional Statements Or Selection Statements	**Loop Control Statements or Iterative Statements**	**Jump Control Instructions or Branching Statement**
• if • if-else • Nested if-else • else if Ladder • switch-case	• for • while • do-while	• break • continue • return

if

- if - is a decision-making statement.
- "if" is the keyword used to decide to control the flow of execution.

Syntax:-

1. **'if' with single statement**

```
if(condition)
        true_statement;
```

- In the above syntax, the condition is evaluated first; if the condition is evaluated as true, then true_statement is executed.
- The default scope of 'if' is a single statement; that's why there is no need to use curly braces.

'if' with multiple statement.

```
if(condition){
        true_statement 1;
        true_statement 2;
        …….
        true_statement n;
}
```

Using curly braces increased(multiple statements) the scope of the 'if' statement.

if-else

- If-else is a decision-making statement.
- The else clause is an extension to the if clause & contains a false part.
- "if" & "else" are keywords used to decide to control the flow of execution.
- The else block should have matching if, otherwise, mismatch else error will occur**.**
- As per syntax, no condition is required with the else block.

Syntax:-

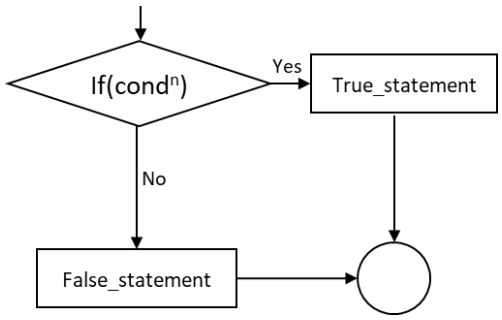
1. **if-else with single statement**

```
if(condition)
        true_statement;

else

        false_statement;
```

Flowchart:-

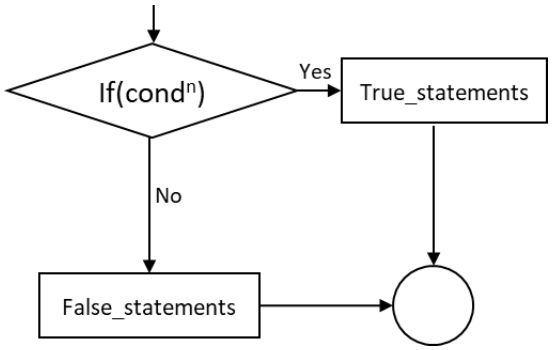


In the above syntax, if (condition) is evaluated first, if the condition is evaluated as true, then true_statement is executed. If the condition is evaluated as false, then false_statement is executed.

2. if-else with multiple statement.

```
if(condition){
        true_statement 1;
        true_statement 2;
        …….
        true_statement n;
}
else{
        true_statement 1;
        true_statement 2;
        …….
        true_statement n;
}
```

Flowchart:-



Using curly braces shows the increased scope of the 'if' & 'else' clauses. If the condition is evaluated as true, then true_statements are executed; otherwise, false_statements are executed.

Nested if-else

- It is a complex decision-making statement if and/or else clause can be nested one inside another (as per syntax of else).
- If clause may have if-else and/or its chain in nested if-else. Similarly, the else clause may have if-else and/or its chain or both if & else clauses have sub if-else clause/block.
- Complex nested if-else (Multiple decision-making) statements may cause problems maintaining the program.

Syntax:-

```
if(condition){
        if(condition){
                statements;
        }
        else{
                statements;
        }
}
else{
        if(condition){
                statements;
        }
        else{
                statements;
        }
}
```

In the above syntax, the inner if and/or else clauses may have if-else clauses in one another. It may confuse; that's why to be careful when nesting if-else. Nested if-else is nothing but chained with one another.

else-if ladder

- It is a common programming construct used to make multiple decisions. Sometimes, we may call this as if-else-if ladder.
- It is different than that of nested if-else & less confusing than that of nested if-else.

Syntax:-

```
if(condition){
        statements:
}
else if(condition){
        statements;
}
else if(condition){
        statements;
}
:
:
else{
        statements;
}
```

In above format, condition is evaluated from top to down.

switch case

- It is a multiple-branching statement.
- It checks for equality, not condition.

Advantages:-

- Easy to use
- Easy to find out errors(if any) & debug.
- Complexity of the program is minimized.

Syntax:

```
switch(equality_constant or variable or expression){
        case constant1:
                statement1;
                break;
        case constant2:
                statement2;
                break;
        :
        :
```

```
        case constant_n:
                statement2;
                break;
        default:
                default_statement;
    }
```

In above syntax:

- equality_constant or variable or expression should be of type int or char.
- default is optional

Nested switch case:

- outer switch block may contain inner switch block, i. e. switch with in a switch.
- The inner and/or outer switch may contain same equality constant.

Syntax:-

```
switch(equality_constant or variable or expression){
        case constant1: statement1; break;
        case constant2: statement2; break;
        :
        case constant_n: statement2;  break;
        default:  default_statement;
                switch(equality_constant or variable or expression){
                        case constant1: statement1; break;
                        case constant2: statement2; break;
                        :
                        case constant_n: statement2;  break;
                        default:  default_statement;
                }

        }
```

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**

BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

Instructions: - All the students are informed to write all executed code in workbook in following sequence and format.

1. Problem Statement
2. Input and Output
3. Test Cases
4. Flowchart
5. Program (with color codes)
   - **a)** Red – Directives
   - **b)** Blue – Keywords, constants values
   - **c)** Green – Comments, messages
   - **d)** Black – {variables, functions, class, object} name, operators, punctuation
6. Trace Table (additional columns may require in some concepts)

Var-1	Var-2	….	Var-n	Condition	Output

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

Tasks:

Sr. No.	Problem Statement	I/O	Test Cases	Flow chart	Program-with color codes	Trace Table
1	Develop a program that accepts sales amount; if the sales amount is more than 5000, then the discount is 12% of the sales amount; otherwise, it is 7%—display the total discount and amount to be paid after the discount.	✓		✓	✓	
2	Write a program to test whether a given character is a capital or small letter and change small letters to capital letters and vice versa.	✓		✓	✓	
3	Implement a program to accept a year as input and print whether it is a leap. A year is a leap if divisible by 4, and centennial years (years divisible by 100) are leap years only when divisible by 400.	✓	✓	✓	✓	✓
4	Develop a program to perform divisibility tests by 3 and 5. If the entered number is divisible by three and not by five print "THREE"; if the number is divisible by five and not by three print "FIVE"; if divisible by both 3 & 5 print "BOTH" otherwise, print "NOT"	✓		✓	✓	
5	2i **D3 in nm/ML** / **Status** <20 / Deficiency 20-30 / Insufficiency 30-100 / Sufficiency >100 / Toxicity	✓	✓	✓	✓	
6	Write a program that takes three coefficients (a,b,and c) of a quadratic equation; $ax^2+bx+c=0$ as input, compute all possible roots, and print them with appropriate messages.	✓	✓	✓	✓	✓

7		✓			✓	
8	Write a program that takes an arithmetic operator (+, -, *, or /) and two operands from the user. Perform corresponding arithmetic operations on the operands using switch case.	✓	✓	✓	✓	
9	Implement a menu-driven program to calculate the area of a triangle, rectangle, circle, and sphere.	✓			✓	

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**

	SVKM's NMIMS **Mukesh Patel School of Technology Management & Engineering / School of Technology Management & Engineering**	
BTI	**Workbook**	**Academic Year- 2023-24**
Year: III	**Subject:-** Programming for Problem Solving	**Semester: V**