

ID 15499

Page ①

Exam : Final

Semester : 4th

Subject

Programming

Fundamentals

ID : 15499

Amir Abbas

Teacher

Dr. Fazal-e-Malik

Statement Purpose:-

The various forms of if statements are Fortran, main branching tool. They give Fortran and an ability to make decisions in a program. The different forms of if statements that can be used include the simple logical if, the then-else structure and the arithmetic if.

The if statements:-

Use the if statement a block of C + code be executed if a condition is true.

Example:-

```
if (20 > 18) {  
  cout << "20 is greater than 18";  
}
```

The else if statement:-

Use the if statement to specify a new condition first condition is false.

Page (3)

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int time = 22;
    if (time < 10)
    {
        cout << "Good morning.";
    }
    else if (time > 20)
    {
        cout << "Good day.";
    }
    else
    {
        cout << "Good evening.";
    }
    return 0;
}
```

Ans (15) (b)

```

#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int a, b, big;
    cout << "Enter two number";
    cin >> a >> b;
    if (a > b)
    {
        big = a;
    }
    else
    {
        big = b;
    }
    cout << "Biggest of the two
    number is " << big;
    getch();
}

```

Result :-

Enter two number : 3

4

Biggest of the two number is 4

Ans // 3 (a)

meaning :-

Loop structure allow us to execute a statement or a block of statement repeatedly.

Types :-

- 1 while loop
- 2 for loop
- 3 Do while loop

while loop :-

A while loop statement in C++ programming language repeatedly executes a target statement as long as a give condition is true.

Syntax

while (condition)

{

Statements

}

Here statement may be a single or a block of statement.

For loop:-

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to be executed a specific number of times. A for loop is useful when you know how many times a task is to be repeated.

Syntax

```

For (initialization, condition,
increments)
statement
}

```

Here is flow of control

1. The initial step is executed 1st and only once.

2. The condition is evaluated 1st and if it is true the body of the loop is executed or if it is false the body of the loop will not be executed and control jump to the next statements past the for loop.

Do while loop:-

It is similar to while loop except that a do while loop is guaranteed to execute at least one time

Syntax

Do

{ statement

} while (boolean - expression);

not the boolean expression appears at the end of the loop so the statements in the boolean is tested

Nested loops:-

nested loop within the body of another loop is known as a nested loop. Syntax is like a do while loop. It runs double time.

Ans(3)(b)

```
#include <iostream>
using namespace std;
```

```
int main()
```

```
{
```

```
    int n;
```

```
    cout << "Enter an integer:";
    cin >> n;
```

```
    if (n % 2 == 0)
```

```
        cout << n << " is even";
    else
```

```
        cout << n << " is odd";
    return 0;
```

```
}
```

~~Result~~ Result :-

Output

Enter an integer: 23
23 is odd.

Ans (4) (a)

The Break statement-

→ There are situations where we want to jump out of a loop instantly without waiting to get back to the condition test.

→ The keyword `break` allows us to do this

→ When `break` is encountered inside any loop, control automatically passes to the first statement the loop.

→ A `break` is usually associated with an `if`.

→ The keyword `break` breaks the control only from the loop in which it is placed.

The continue statement-

→ `continue` statement allows to take the control to the beginning of the loop bypassing the statement inside the loop which have not yet been executed.

A `continue` is usually associated with an `if`.

Ans (4) (b)

```
#include <iostream>
using namespace std;
int Main()
```

```
{
    int i, sum = 0;
```

```
    cout << "The natural numbers are
    : \n";
```

```
    for (i = 1; i <= 10; i++)
```

```
    {
        cout << i << " ";
        sum = sum + i;
```

```
    }
```

```
    cout << "\n The sum of
    first 10 natural
    numbers : " << sum << endl;
```

```
}
```

Result:

The natural number are:
1 2 3 4 5 6 7 8 9 10

The sum of first 10 natural
numbers: 55

Ans (2) (a)

A logical operator is a symbol or word used to connect two or more expressions such that the value of the compound expression produced depends only on that of the original expressions and on the meaning of the operator. Common logical operators include AND, OR, and NOT.

Logical operators.

and (logical AND)

- Used to combine two conditions
- true if both conditions are true

```
if (gender = 1 and age >= 65)
```

• || (logical OR)

- true if either of conditions is true

```
if (semester Avg >= 90 || final Exam >= 90)
    cout << ("student grade is A");
```

! (logical NOT)

! (logical NOT, logical negation)

- Returns true when condition is false & vice versa

```
if (! (grade = 20))
    cout << "hello world";
```

Alternative:

```
if (grade != 20)
    cout << "hello world";
```

Ans (2) (b)

```
#include <iostream>
using namespace std;
main()
```

{

```
int temperature;
```

```
cout << "enter the temperature" << endl;
```

```
cin >> temperature;
```

```
if (temperature > 40)
```

```
{
```

```
    cout << "very hot";
```

```
    else if (temperature > 35 and temperature <
= 40)
```

```
{
```

```
    cout << "tolerable";
```

```
}
```

```
    else if (temperature >= 30 and temperature <=
35)
```

```
{
```

```
    cout << "warm";
```

```
}
```

```
    else
```

```
    cout << "cool";
```

```
}
```

Result

Enter the temperature
39
tolerable

Process exited after 5.17 seconds
with return value 0

Process any key to continue---

Ans (5) (a)

In C++, character set is of all valid character that be used in a C++ program character set is used to specify the characters or symbols recognized by the language that character set is of all valid characters that can be used to form word numbers and expressions in source programs.

Letters: A-Z, a-z

The 26 lowercase Roman characters:-

~~a b c d e f g h i j k l m n o p q r s t u v~~
 a b c d e f g h i j k l m n o p q r s t u v
 x y z

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Digist: 0-9

The 10 decimal digits:-

0 1 2 3 4 5 6 7 8 9

Special symbols :- Space + - * / () [] { } = ! < >
 ; : % | & - # < > =

The graphic character:

samp + () - = + ~ ' " : ; ? / \ ()
[] ' -

White Spaces:- Blank space, Horizontal tab
(-) carriage return (-)
Newline, form feed.

specs () Horizontal tab (ii) carriage return
(iv) New line (v) form feed (v)

Other characters C++ can process any of
256 ASCII character as data or
as literals:

Example:-

```
#include <iostream>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
    char letter, digit, special, white;  
    cout << "Enter w3Adda - C++ character set  
example" << endl;
```

```
    cout << "Enter a letter:";
```

```
    cin >> letter;
```

```
    cout << "you entered a letter " << letter <<  
    " " << "\n";
```

```
    cout << "Enter a Digit, :";
```

```
    cin >> digit;
```

```
    cout << "you entered a digit " << digit <<  
    " " << "\n";
```

```
cout << "Enter a special character: ";  
cin >> special;  
cout << "You entered a special character  
" << special << " " << endl;  
cout << "A horizontal (\t) tab";  
  
return 0;  
}
```

Out put:-

W3Add - C++ character set Example
Enter a letter : A
You entered a letter 'A'
Enter a digit : 5
You entered a digit '5'
Enter a special character : 0
You entered a special character '0'
A horizontal () tab

Ans (5) (b)

Constant Definition in C++

There are two other different ways to define constants in C++, these are

- By using const keyword
- By using ~~#define~~ preprocessor.

Constant Definition by using const keyword

Example:

```
#include <iostream>
using namespace std;
```

```
int main()
```

```
{
```

```
    const int SIDE = 50;
    int area;
```

```
{
```

```
    const int SIDE;
```

```
    cout << "The area of the square with side: " << SIDE << " is: " << area << endl;
```

```
    system ("PAUSE");
```

```
    return 0;
```

```
}
```

Program output:-
 The area of the square with side
 so is 2500
 Press any key to continue ----

constant Definition using #define Preprocessor

Example:-

```
#include <iostream>
```

```
using namespace std;
```

```
#define VAL1 20
```

```
#define VAL2 6
```

```
#define Newline '\n'
```

```
int main()
```

```
{
```

```
    int tot;
```

```
    tot = VAL1 + VAL2;
```

```
    cout << tot;
```

```
    cout << Newline;
```

```
}
```

Ans (5) (c)

Variable Definition in C++

A variable definition tells the compiler where and how much storage to create for the variable. A variable definition specifies a data type, and contains a list of one or more variable of that type as follows -

```
type variable - list;
```

Here, type must be a valid C++ data type including char, w, char, int, float, double, bool or any user-defined object - etc, and variable list may consist of one more identity names separated by commas some valid declarations are shown here.

```
int i, j, k;
char c, ch;
float f, Salary;
double d;
```

The line `int i, j, k;` both declares and defines the variable `i, j` and `k`, which instruct the compiler to create variable named `i, j` and `k` type `int`

follows consists of an equal sign followed by an expression as

Types of Variable: - name = value;

Some examples are -

```
extern int d = 3, f = 5; // declaration of d and f.
int d = 3, f = 5; // declaration and initialization of d and f.
byte z = 22; // declaration and initialization of z.
char x = 'x'; // declaration x has the value 'x'
```

Ans (5) (D)

Keywords In C++

Keywords is predefined or reserved word in C++ library a fixed meaning and used to perform an operation C++ language supports more than 54 keyword.

keyword are those words whose meaning is already defined by compiler these keyword cannot be used as an identifier note that keyword are the collection of reserved word and predefined word and identifiers that are defined by the compiler but can be changed meaning by user:

Every keyword exists in lower case latter auto break case continue etc.

Key word in C++ Language.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sized	volatile
do	if	static	while

```
#include <iostream>
using namespace std;
int main ()
{
    int n;
    cout << "Enter number: " << endl;
    cin >> n;
    if (n > 0)
    {
        cout << "you have entered positive
        number";
    }
    return 0;
}
```

Result

Enter number:

10

You have entered positive number.

Ans (5) (E)

C++ Relation operators:-

In C++ programming the values stored in two variables be compared using following operators and between them can be determined.

There are six relational operators
They are,

① < less than

② > greater than

③ <= less than or equal to

④ >= greater than or equal to

⑤ == equal to

⑥ != not equal to

Example of Relational operator program

```
// Header files
```

```
#include <iostream>
```

```
#include <conio.h>
```

```
// Main function
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
    // variable Declaration
```

```
    int a = 25, b = 5;
```

```
    // int a = 5, b = 25;
```

```
    // int a = 5, b = 5;
```

```

cout << "Simple Relational operator
Example program \n";
if (a > b)
    cout << "A is Sig";
else if (a == b)
    cout << "A and B are Equal";
else
    cout << "B is Big";

// wait for output screen
getch();
// Main function return statement
return 0;
}

```

Output

```

Simple Relation operator example program
// if a = 25, b = 5;
A is Big

```

```

// if a = 5, b = 25;
B is Big

```

```

// if a = 5, b = 5;
B is Big

```