Name : Hamza Iqbal

ID : 14784

Teacher : Dr. Fazale-Malik

Subject : Programming Fundalmentals

Program : BS (SE)

Q No 2(a)

What are the Logical Operators? Explain them.

Answer:

Logical Operators:

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 AND)

-Used to combine two conditions.

-true if both condition are true

```
if (gender == 1 && age >= 65)
    senior++;
```

- || (logical OR):

- true if either of condition is true

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

- **!(Logical NOT)**
- Return true when its condition is false, & vice versa

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

Alternative:

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

Q No 2:

(b) write a c++ program to get Temperature ----- ?

Answer:

```
#include<iostream>
#include<conio.h>
using name-space std;
main()
{
    int temp;
    cout<<" temperature is\n";
    cin>>temp;
  if (tmp>=40)
   {
     cout<<" its very hot\n";
   }
    else if (tmp>35 && temp<40)
   {
    cout<<" its tolerable\n";
   }
    else if (tmp>=30 && tmp<=35)
   {
    cout<<" its warm\n";
   }
    else
   {
    cout<<  "cool";
    }
}
```

Q No 1:

(a) What is the purpose of if statement? Dissue its two different forms with examples.

Answers:

if Statement.

An if statement is a programming conditional statement that, if proved true, performs a function or display information. Below is a general example of an if statement, not specific to any particular programming language.

```
if (x < 10) {
    print " Hello world";
}
```

In the example above, if the value of x were equal to any number Lees than 10, the program display "Hello world";

Two form of if statement

Two types of if condition are,

(i) if-Then

(ii) If-Then else

If Condition:

```
#include <iostream>
using namespace std;
main()
{
    if (condition)
    {
        // Body of if statement
    }
}
```

If-Then-Else condition:

```
main()
{
    if (condition)
    {
        // Block of code if condition is true
    }
    else
    {
        // block of code if condition is false
    }
}
```

Q.No1 :

(b) Write a c++ program to read two number from keyboard and then find the Largest number of them.

Answers :—

```
#include <iostream>

using namespace std;
int main()
{
    int n1 , n2;
    cout << "Enter two number from keyboard:
          \n";
    cout << " enter first number \n";
    cin >> n1;
    cout<<" enter 2nd number \n";
    cin >> n2;

  if( n1 >= n2)
{
  cout<< "Largest number is :) " ;
  cout<< n1;
}
else {
     cout << " Largest number is :";
     cout << n2;
}
     return 0;
}
```

Date:

Q No 5:

Explain the following with proper examples.

(b) Constants:

Constants refer to immutable value. Constants are basically literals whose value cannot be modified or change during the execution of program. Constant are also called as literals. constant must be initialized when declared as value cannot be assigned it to letter.

Constants can be following four basic types

(1) Integer constant
(2) Floating point constant
(3) Character constant
(4) String contant

Example of constant:

Constant can be created in following two ways.

* Using # preprocessor
* using const keywords.

```
#include <iostream>
using name space std;
#define lenght 10
#define width 5
#define Newline '\n'
```

```
int main()
{
    int area;
    area = length * width
    cout << area;
    cout << Newline;
    return 0;
}
```

(c) Variables:

Variable is an identifier used to refers memory located in computer memory that hold a value for that variable, this value can be changed during the execution of program. When you create a variable in C++, this means you are allocating some space in the memory block allocated and types of the value it holds is completely dependent upon the types of variable.

Syntax:

<types> < variable-name >

types name of variable.

(i) Declaring variable
(ii) Declaring multiple variable
(iii) Variable assignment etc.

(a) C++ character set.

In C++ Charater set is a set of all valid character that can be used in a C++ program. Character set can be used to specify the character or symbols recognized by the language. character set is a set of all valid character that can be used to form words, number and expression in source program.

Letters:

A to Z, a-Z

Digits:

The 10 decimal digits

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Special Symbol:

Space + - * / ^ > < ( ) [ ] { } = != , . " & # etc

(d) Keywords:

In C++, there are is a set of reserved word that you cannot use as an identifier. These words are know as "reserved word" or "keyword". Keywords are standard identifer and their function of predefined by the

M T W T F S S

Compiler. We cannot use keyword as variable name, class name, or method name etc.

(e) Relational operators:

A relational operator is used to check the relationship b/w two operands. for example

```
// check if a is greator than b
a > b;
```

Here, > is a relational operator. It check if a is great than b or not

If the relation is true, it return 1, if the relation is false, it return 0.

Q No 3:

(a) What does Looping mean? Explain different Loop in C++.

Answer:

A Loop is used for executing a block of statement repeatedly until a particular condition a satisfied. For example when are you may want set of value of a vaviable to 1 and display it 100 times increasing its value by 1 on each loop iteration.

Types of Loop in C++

In C++ three types of basic Loops.

(i) for Loop
(ii) while loop
(iii) do-while loop

(i) For Loop:

for loop is used to execute a set of statement repeatedly until a particular condition is satistied. It an open ended loop.

for (initialization; condition; increment/decrement) {statement-block;}

Date:

In for loop we have exactly two semicolons, one after initialization and second is after condition.

(ii) **While Loop:**

While loop can be address an entry control loop. It completed in 3 steps.

Variable initialization. (e.g int n = 0)
condition (e.g while(n <= 10)
Variable increament or decrement
(n++ or n-- or n = n+2)

Syntax:

Variable initiatiation; while (condition) { statement; variable increment or decrement; }.

(iii) **do-while Loop:**

In some situation it is necessary to execute body of the loop before testing to condition. Such situation can be handled with the help of do- while loop. Do statement evaluation the body of the loop first and at the end, the condition is checked.

Using while statement. General format of dowhile is, do {// a couple of statements}
while (condition);

Q No 3:

(b) Write a c++ program to read a number - - - - -- even or odd number?

Answer

```
#include <iostream>
#include <conio.h>
using namespace std;
main()
{
    int number;
    cout<<" enter the number\n";
    cin >> number;
    if (number % 2 == 0)
        cout << number <<" is an even number";

    else
        cout <<number <<" is an odd number";
        return 0;
}
```

Q No 4:

(b) Write a C++ program to find the sum of following numbers

1 + 2 + 3 + - - - + 10.

```
#include <iostream>
#include <conio.h>
using namespace std;
int main()
{
    int i;
    int sum;

    cout<<"The first 10 natural number are:\n";
    for(i=1; i<=10; i++)

    {

    cout<< i<<" ";
    sum = sum+1;
    }
    cout<<"\n The sum of first 10 natural
                    number is \n";
    cout<< sum;
    return;
}
```

DAWN Imported
M T W T F S S

Date:

Q No 4:

(a) What is the purpose of using break and continue statement.

Answer:-

Break Statement:

Break statement to come out of the Loop instantly. whenever a break statement is encountered inside a loop, the control directly comes out of loop terminating it. It is used along with if statement whenever used inside loop so it occurs only for a particular condition.

Syntax of break statement

```
break;
```

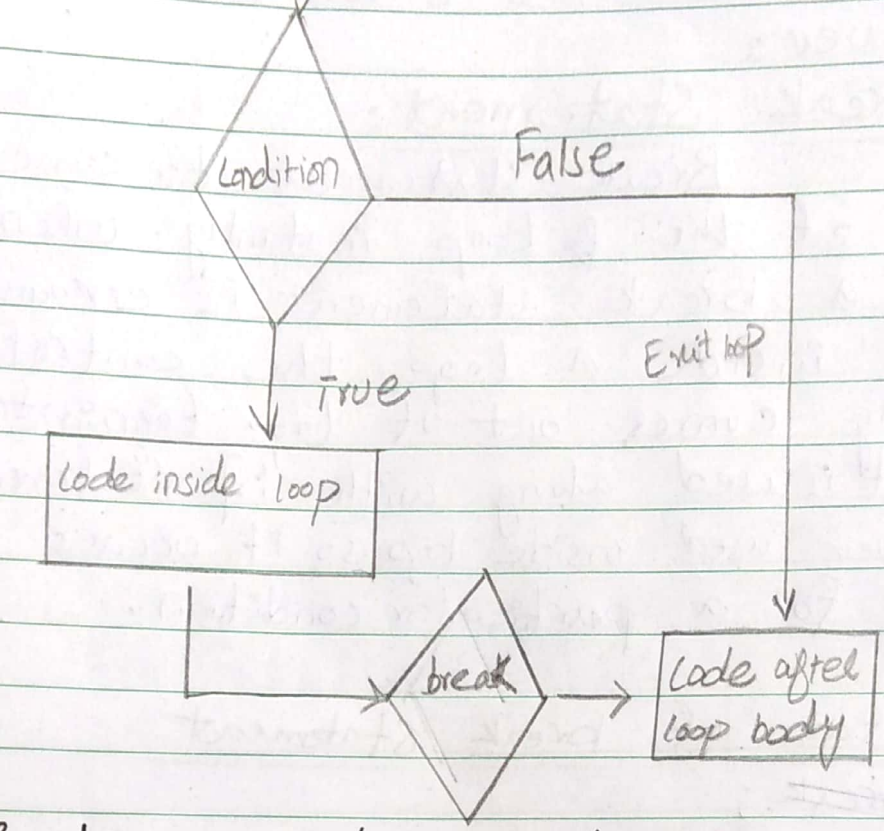
Continue Statement

The continue statement is used inside loops. when a continue statement is encountered inside a loop, control jumps to beginning of the loop for next iteration, skipping the execution of statement inside the body of loop for the current iteration.

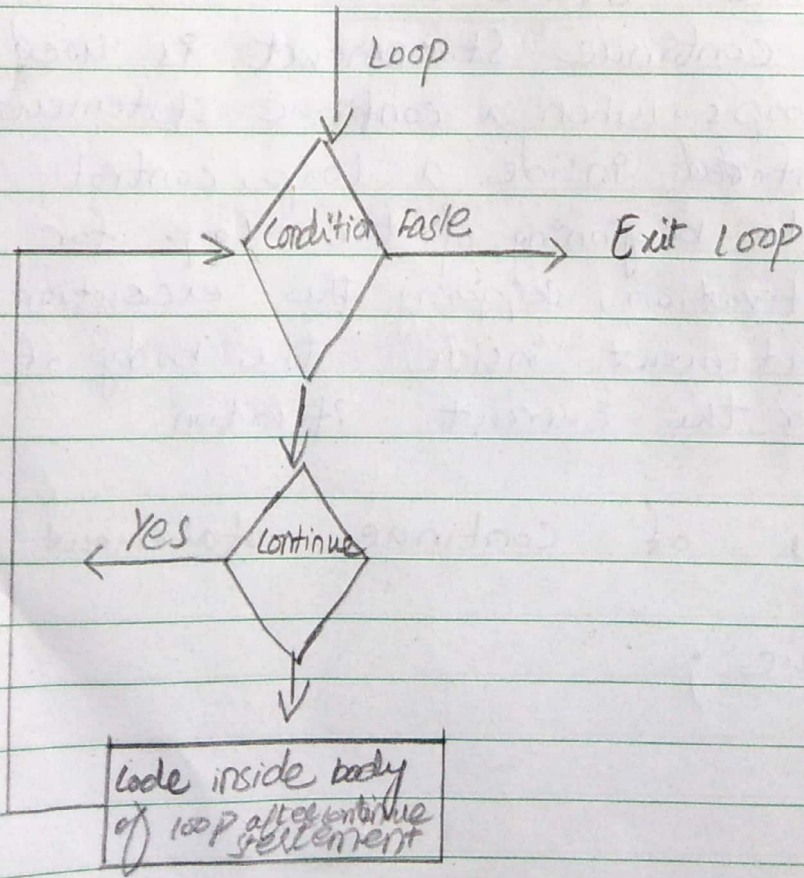
syntax of continue statement

```
continue;
```

Break statement



Continue statement



Different b/w break and continue statement:

Break and continue are same types of statement which is specifically used to alter the normal flow of the program still they have same difference b/w them.

Break Statement:

The break statement terminates the smallest enclosing loop (i.e while loop, do-while, for or switch statement.

Continue Statement:

The continue statement skips the rest of loop statement and causes the iteration of the loop to take place.