

(1)

Ayesha Mehmood

ID # 6832

Subject # Programming

Fundamental

Degree # BSCS

Semester # 8th

Final-term

Date # 26 - Jun - 2020

Question # 3 (a)

Answer

Looping:-

A loop is used for executing a block of statements repeatedly until a particular condition is satisfied. In C++ we have three types of basic loops.

(2)

- For loop
- while loop
- do-while loop

For loop

Execute a sequence of statements multiple times
And abbreviates that code that manages the loop variable.

While loop

Repeat a statement or group of statements while a given condition is true.
It tests the condition before executing the loop body.

Do-while loop

Like a while statement except that it tests the condition at the end of the loop body.

(3)

Question # 4 (a)

Answer

Break Statement :-

When a break statement is executed, the most deeply nested loop currently being executed is ended and execution picks up with the next statement after the loop.

Continue Statement :-

The continue statement ends that current operation of the loop and returns to the condition at the top of the loop. Such loops are typically used to exclude some values from calculations.

(4)

Question # 05

Answer

Array:-

- Offers a simple way of grouping like variables for easy access.
- It is a group of elements having same data type.
- An array is a collective name given to a group of similar quantities.
- Arrays in C share a few common attributes.
- Variables in an array share the same name.

(5)

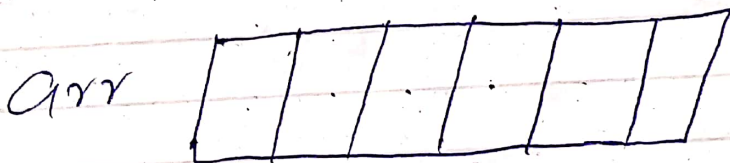
- Individual variables in an array are called elements.

- Elements in an array are accessed with an index number.

Types of Array:-

One - Dimensional:-

Conceptually you can think of an one dimensional array as a row, where elements are stored one another.



Example:-

```
#include <stdio.h>
void odd-or-even (int a);
```

(6)

```
int main()
```

```
{
```

```
int my_arr[] = {13, 56, 71, 38,  
                93};
```

```
for (i=0; i<5; i++)
```

```
{
```

```
// Passing one element at a  
time to add-or-even()  
function odd-or-even(my_arr  
[i]);
```

```
}
```

```
// signal to operating system  
Program run time
```

```
return 0;
```

```
}
```

```
void odd-or-even(int a)
```

```
{
```

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

(7)

```
{ print f ("%d is even/n", a);  
}  
}  
else  
{  
Print f ("%d is odd/n", a)  
}  
}
```

Two-Dimensional:

An array of arrays is known as 2D arrays. The two dimensional array in C programming is also known as matrix. A matrix can be represented as table of rows and column.

Example:-

```
#include <stdio.h>
int main() {
```

```
int disp[2][3];
```

```
int i, j;
```

```
for (i=0; i<2; i++) {
```

```
for (j=0; j<3; j++) {
```

```
Print { "Enter value for disp  
[%.d][%.d];"
```

```
scanf("%.d", disp[i][j]);
```

```
}
```

```
}
```

```
Print ("Two Dimensional Array  
elements.\n");
```

```
for (i=0; i<2; i++) {
```

```
for (j=0; j<3; j++) {
```


(9)

```
printf("70d", disp(i][j]);
```

```
if (j == 2) {
```

```
    printf("\n");
```

```
}
```

```
}
```

```
}
```

```
return 0;
```

```
}
```

Question # 02(9)

Answer:-

Logical Operators:-

A logical operators is a symbol or word used to connect two or more expression such

(10)

that the value of the compound expression produced depends only on that of the original expression and on the meaning of the operator. Common logical operators include AND, OR and NOT.

Operator

Description

& &

called Logical AND operator, if both the operands are non-zero, then condition become true.

||

Called Logical OR operator. If any of the two operands is non-zero, then condition becomes true.

|

Called Logical NOT operator. Use to

(11)

reserves the logical state of its operand. If a condition is true, the logical NOT operator will make false.

Question # 01 (a)

Answer

IF Statement Purpose

The various forms of if statements are fortran's main branching tool - they give fortran an ability to make decision in a program. The different forms of if statement can be used include the simple logical if, the if-then-else structure, and the arithmetic if.

(12)

IF then-else Statement:-

Example:-

```
if (delta >= 1) then
    print *, 'Too much deflection'
else if (shear >= 21) then
    print *, 'Beam fails in shear'
else if (sigma >= 36) then
    print *, 'Beam fails in tension'
else if (sigma <= -36) then
    print *, 'Beam fails in compression'
else
    print *, 'Beam will not fail
    under these & conditions'
end if.
```

IF - else statements.

else - If statement:-

The else - if statement is useful when you need to check multiple conditions with in the program, nesting of if - else blocks can be avoided using else - if statement.

Example of else - if statement:-

```

#include <stdio.h>
int main ()
{
    int var1, var2;
    printf ("input the value of var1:");
    scanf ("%d", &var1);
    printf ("input the value of var2:");
    scanf ("%d", &var2)

    {
        printf ("var1 is not equal to
                var2\n");
    }
}

```

(14)

```
else if (var2 > var1)
```

```
{
```

```
    print ("var1 is greater than  
          var2\n");
```

```
}
```

```
else if (var2 > var1)
```

```
{
```

```
    print ("var2 is greater than  
          var1\n");
```

```
}
```

```
else
```

```
{
```

```
    print ("var1 is equal to var2\n");
```

```
}
```

```
    return 0;
```

```
}
```

15

Question # 01
(b)

Answer

```
#include <iostream>
```

```
int main() {
```

```
    int a;
```

```
    int b;
```

```
    cout << "enter two  
    numbers:";
```

```
    cin >> a >> b;
```

```
    if (a > b)
```

```
        cout << "a is greater";
```

```
    else
```

```
        cout << "b is greater"; }
```

(16)

Question # 02
(b)

Answer

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
int f;
```

```
cout << "enter the F: ";
```

```
cin >> f;
```

```
if (f < 30) {
```

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

```
else if (f > 30) {
```

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

```
else if (f > 35)
```

```
{
```


17

{

cout << "tolerate"; }

else if (f > 40)

{

cout << "very hot"; }

Question # 03
(b)

Answer

```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
int n;
```

```
cout << "Enter integer: ";
```

```
cin >> n;
```

(18)

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

```
    cout << "the number is even.";
```

```
else
```

```
    cout << "the number is odd.";
```

```
}
```

Question # 04

(b)

Answer

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

```
{
```

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

```
    cout << "\n\n find the first 10
    natural numbers : \n";
```

```
    cout << "\n\n find the first 10
    natural numbers : \n";
```

(19)

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;

}