

Name	Symbol
a) Modulus operator	i) ++
b) Logical operator	ii) ==
c) Relational operator	iii) =
d) Assignment operator	iv) ?:
e) Increment operator	v) &&
f) Conditional operator	vi) %

**Ans.** a) vi      b) v      c) ii      d) iii      e) i      f) iv

\*\*\*\*\*

## Chapter 7 – Control statements

1. Compare entry controlled loop and exit controlled loop.

[ March 2020, Score 2 ]

**Ans.**

Entry controlled loop	Exit controlled loop
Condition is checked before the execution of the body	Condition is checked after the execution of the body
No guarantee to execute the loop body at least once	Will execute the loop body at least once even though the condition is False
Suitable when skipping of the body from being executed is required	Suitable when normal execution of the body is to be ensured.

2. Write the syntax of 'for' statement used in C++.

[ March 2020, Score 2 ]

**Ans.**

for (initialisation; test expression; update statement)

```
{  
body-of-the-loop;  
}
```

3. (a) Differentiate 'break' and 'continue' statement in table form [ March 2020, Score 2 ]

(b) Write the symbol and use of conditional operator. [ March 2020, Score 2 ]

(c) Write the example of nested if statement. [ March 2020, Score 1 ]

**Ans.**

a)

break statement	continue statement
Used with switch and loops.	Used only with loops.
Brings the program control outside the switch or loop by skipping the rest of the statements within the block.	Brings the program control to the beginning of the loop by skipping the rest of the statements within the block.
Program control goes out of the loop even though the test expression is True.	Program control goes out of the loop only when the test expression becomes False.

b) ?: is the conditional operator. It can be used as an alternative to if...else statement.

Its general form is:

Test expression ? True\_case code : False\_case code;

c) if (score >= 60)

```
{  
if ( age >= 18 )  
    cout<< " You are selected for the course! " ;  
}
```

4. Write the use and syntax of the following operators in C++

(a) Conditional operator ( ? : )

(b) Size of operator

[ July 2019, Score 2 ]

**Ans.**

a) ?: is the conditional operator. It can be used as an alternative to if...else statement.

Its general form is:

Test expression ? True\_case code : False\_case code;

b) The *sizeof* is a keyword that determines the *size*, in bytes, of a variable or data type

Syntax is : sizeof (data type)

5. Compare for loop and do-while loop in C++.

[ July 2019, Score 2 ]

**Ans.**

for loop	do while loop
Entry controlled loop	Exit controlled loop
Initialisation along with loop definition	Initialisation before loop definition
No guarantee to execute the loop body at least once	Will execute the loop body at least once even though the condition is False

6. Explain about the five jump statements in C++

[ July 2019, Score 5 ]

**Ans.** The statements that facilitate the transfer of program control from one place to another are called jump statements.

1. goto statement

The goto statement can transfer the program control to anywhere in the function. The target destination of a goto statement is marked by a label, which is an identifier.

The syntax of goto statement is:

```
goto label;
```

```
.....;
```

```
.....;
```

```
label: .....;
```

```
.....;
```

2. break statement

When a break statement is encountered in a program, it takes the program control outside the immediate enclosing loop ( for , while , do...while ) or switch statement

E.g while(i<=10)

```
{cin>>num;
```

```
if (num==0)
```

```
break;
```

```
-----
```

```
-----}
```

3. continue statement

continue statement is another jump statement used for skipping over a part of the code within the loop-body and forcing the next iteration.

E.g. for (i=1; i<=10; ++i)

```
{
```

```

if (i==6)
continue;
cout<<i<<"\t";
}

```

#### 4. return statement

The return statement is used to transfer control back to the calling program or to come out of a function.

E.g. return 0;

#### 5. exit()

Built in function to terminate the program itself. To use exit( ), the header file `stdlib` to be included.

7. Compare Entry Controlled loop with Exit Controlled loop.

[ March 2019, Score 2 ]

**Ans.**

Entry controlled loop	Exit controlled loop
Condition is checked before the execution of the body	Condition is checked after the execution of the body
No guarantee to execute the loop body at least once	Will execute the loop body at least once even though the condition is False
Suitable when skipping of the body from being executed is required	Suitable when normal execution of the body is to be ensured.

8. Write the syntax of if else statement.

[ March 2019, Score 2 ]

**Ans.** if (test expression)

```

{
    statement block 1;
}
else
{
    statement block 2;
}

```

9. a) Explain any three jump statements with examples

[ March 2019, Score 3 ]

b) Read the following C++ statement and rewrite using if – else

`(a>b) ? cout<<a : cout <<b;`

[ March 2019, Score 2 ]

**Ans.** a) The statements that facilitate the transfer of program control from one place to another are called jump statements.

### 1. goto statement

The goto statement can transfer the program control to anywhere in the function. The target destination of a goto statement is marked by a label, which is an identifier.

The syntax of goto statement is:

```
goto label;  
.....;  
.....;  
label: .....;  
.....;
```

### 2. break statement

When a break statement is encountered in a program, it takes the program control outside the immediate enclosing loop ( for , while , do...while ) or switch statement

E.g while(i<=10)

```
{  
cin>>num;  
if (num==0)  
break;  
----  
----}
```

### 3. continue statement

continue statement is another jump statement used for skipping over a part of the code within the loop-body and forcing the next iteration.

E.g. for (i=1; i<=10; ++i)

```
{  
if (i==6)  
continue;  
cout<<i<<"\t";  
}
```

```
b) if (a>b)  
    cout<<a;  
    else  
    cout<<b;
```

10. .... built in function is used to terminate the program

[ July 2018, Score 1 ]

Ans. exit ( )

11. Explain the looping statements in C++

[ July 2018, Score 3 ]

**Ans.** C++ provides three loop statements: while loop, for loop and do...while loop.

#### while statement

The while loop is an entry-controlled loop. The condition is checked first and if it is found True the body of the loop will be executed. That is the body will be executed as long as the condition is True.

The syntax of while loop is:

```
initialisation of loop control variable;
while(test expression)
{
    body of the loop;
    updation of loop control variable;
}
```

#### for statement

The for loop is also an entry-controlled loop in C++. All the three loop elements (initialisation, test expression and update statement) are placed together in for statement

The syntax is:

```
for (initialisation; test expression; update statement)
{
    body-of-the-loop;
}
```

#### do...while statement

In the case of for loop and while loop, the test expression is evaluated before executing the body of the loop. If the test expression evaluates to False for the first time itself, the body is never executed. But in some situations, it is necessary to execute the loop body at least once, without considering the result of the test expression. In that case the do...while loop is the best choice.

Its syntax is :

```
initialisation of loop control variable;
do
{
    body of the loop;
    Updation of loop control variable;
} while(test expression);
```

**12.** Write a program to check whether the given number is prime or not. [ July 2018, Score 5 ]

**Ans.**

```
int main()
```

```

{
  int n, i, flag = 0;
  cout<<"Enter a number";
  cin>>n;
  for(i=2; i<=n/2; ++i)
  {
    if(n%i==0)
    {
      flag=1;
      break;
    }
  }
  if (flag==0)
    cout<<n<<" is a prime number";
  else
    cout<<n<<" is not a prime number";
  return 0;
}

```

13. Which statement in C++ can transfer control of a program to a labelled statement?

[ March 2018, Score 1 ]

Ans. goto

14. Distinguish between Entry- Controlled loop and Exit - Controlled loop.[ March 2018, Score 3 ]

Ans.

Entry controlled loop	Exit controlled loop
Condition is checked before the execution of the body	Condition is checked after the execution of the body
No guarantee to execute the loop body at least once	Will execute the loop body at least once even though the condition is False
Suitable when skipping of the body from being executed is required	Suitable when normal execution of the body is to be ensured.

15. Distinguish between break and continue statements in C++

[ March 2018, Score 3 ]

Ans.

<b>break statement</b>	<b>continue statement</b>
Used with switch and loops.	Used only with loops.
Brings the program control outside the switch or loop by skipping the rest of the statements within the block.	Brings the program control to the beginning of the loop by skipping the rest of the statements within the block.
Program control goes out of the loop even though the test expression is True.	Program control goes out of the loop only when the test expression becomes False.

**16.**

a) What will be the output of the given C++ program? Justify your answer [ March 2018, Score 2 ]

```
#include<iostream>
using namespace std;
int main()
{
int a,b,num;
for ( a = 1 ; a <= 10 ; ++a )
{
for ( b = 1 ; b <= 5 ; ++b )
num = a * b ;
cout<<num<<endl;
}
return 0;
}
```

b) Rewrite the following program using while and do while loops (Write separate programs)

[ March 2018, Score 3 ]

```
#include<iostream>
using namespace std;
int main( )
{ int i ;
for ( i = 1 ; i <= 10 ; i++ )
{
cout<<i ;
}
```



```
return 0;
}
```

**Ans.**

a) 5  
10  
15  
20  
25  
30  
35  
40  
45  
50

Since no braces are used with inner for loop, it takes only the first statement followed by it as its loop body.

b)

using while loop

```
#include<iostream>
using namespace std;
int main()
{ int i;
  i=1;
  while ( i <= 10 )
  {
    cout<< i ;
    i++;
  }
  return 0;
}
```

using do while loop

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

```
int main()
{ int I ;
i = 1;
do
{
    cout<<i ;
    i++;
} while ( i <= 10 ) ;
return 0 ;
}
```

17. Which one of the following is NOT an iteration statement in C++?

- a) while      b) continue      c) for      d) do-- while

[ July 2017, Score 1 ]

**Ans.** continue

18. Write an output of the following C++ statements.

```
int p=1;
for ( int n =-5 ; n <= 5 ; n++ )
p = p * n ;
cout<<p ;
```

- a) 25  
b) 0  
c) -25  
d) None of these

[ July 2017, Score 1 ]

**Ans.** b) 0

19. What is the use of jump statements in a program? Explain the different types of jump statements available in C++.

[ July 2017, Score 5 ]

**OR**

Write a C++ program that accepts a list of integers as input and print sum of even numbers in the list

**Ans.**

The statements that facilitate the transfer of program control from one place to another are called jump statements.

#### 1. goto statement

The goto statement can transfer the program control to anywhere in the function. The target destination of a goto statement is marked by a label, which is an identifier.

The syntax of goto statement is:

```
goto label;
```

```
.....;
.....;
label: .....;
.....;
```

## 2. break statement

When a break statement is encountered in a program, it takes the program control outside the immediate enclosing loop ( for , while , do...while ) or switch statement

E.g while ( i < = 10 )

```
{
cin>>num ;
if ( num == 0 )
break ;
----
----
}
```

## 3. continue statement

continue statement is another jump statement used for skipping over a part of the code within the loop-body and forcing the next iteration.

E.g. for ( i =1 ; i < = 10 ; ++i )

```
{
if ( i == 6 )
continue ;
cout<<i<<"\t ";
}
```

**OR**

```
#include <iostream>
using namespace std;
int main()
{
    int n, sum =0, i ;
    cout<<"Enter numbers" ;
    for ( i = 1; i < = 10 ; i++ )
    {
        cin>>n ;
        if ( n % 2 ==0 )
```

```

        sum = sum + n ;
    else
        continue ;
}
cout<<"The sum is"<<sum;
}

```

**20. a)** Name the type of loop which can be used to ensure that the body of the loop will surely be executed at least once

[ March 2017, Score 1 ]

b) Consider the code given below and predict the output.

```

for ( int i = 1 ; i <= 9 ; i = i + 2 )
{
    if ( i == 5 ) continue;
    cout<<i<<" " ;
}

```

[ March 2017, Score 1 ]

**Ans. a)** do while

b) 1 3 7 9

**21.** Write a program to check whether the given number is palindrome or not

**OR**

Write program to print the leap years between 2000 and 3000

[ March 2017, Score 5 ]

**Ans. a)**

```

int main()
{
    int n, num, digit, rev = 0;
    cout << "Enter a positive number: ";
    cin >> num ;
    n = num ;
    do
    {
        digit = num % 10 ;
        rev = (rev * 10) + digit ;
        num = num / 10;
    } while (num != 0) ;
    cout << " The reverse of the number is: " << rev << endl ;
    if (n == rev)

```

```

    cout << " The number is a palindrome." ;
else
    cout << " The number is not a palindrome. ";

return 0;
}

```

**OR**

```

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

    for (i = 2000 ; i <= 3000 ;i++)
        if ( ( i % 400 ==0 ) || ( ( i % 100 != 0 ) && ( i % 4 == 0 ) ) )
            {
                cout<<i<<" " ;
            }
}

```

**22.** Which one of the following C++ commands transfer the program control to labelled statement?

a) for            b) while            c) break            d) goto            **[ July 2016, Score 1 ]**

**Ans.** goto

**23.** Differentiate between break and continue statements in C++.

**[ July 2016, Score 2 ]**

**Ans.** When a break statement is encountered in a program, it takes the program control outside the immediate enclosing loop ( for , while , do...while ) or switch statement  
 continue statement is another jump statement used for skipping over a part of the code within the loop-body and forcing the next iteration.

**24.** Write C++ program for getting the following output

```

1
1 2
1 2 3
1 2 3 4

```

**[ July 2016, Score 3 ]**

**OR**

Consider the following C++ program and answer the following questions.

```

#include <iostream>

```

```
main()
{
int a, p=1 ;
for( a = 1; a <= 5 ; a += 2 )
p= p * a ;
cout<<p ;
}
```

a) Predict the output of the above code

[ July 2016, Score 1 ]

b) Rewrite the above program using while loop

[ July 2016, Score 2 ]

**Ans.**

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

```
int main()
```

```
{
    int i, j ;
    for ( i = 1 ; i <= 4 ; i++ )
        {
            for ( j = 1 ; j <= i ; j++ )
                cout<<j<<" \t " ;

            cout<<" \n " ;
        }
}
```

**OR**

a) 15

b) #include <iostream>

```
main()
```

```
{
    int a, p=1 ;
    a=1;
    while ( a <= 5 )
    {
        p = p * a ;
        a+=2;
    }
}
```

```
        cout<<p;
    }
}
```

25. How many times the following loop will execute ?

[ March 2016, Score 1 ]

```
int m=2 ;
do
{
cout<<" Welcome " ;
m++ ;
} while(m>10) ;
```

**Ans.** Only once

26. Consider the following statements in C++

[ March 2016, Score 1 ]

```
if ( mark >= 18 )
    cout<<"Passed" ;
else
    cout<<"Failed" ;
```

Suggest an operator in C++ using which the same output can be produced

**Ans.** Conditional operator ( ? : )

27. Write a C++ program to accept an integer number and check whether it is an Armstrong number or not. ( Hint : sum of the cubes of the digits of an Armstrong number is equal to the number itself)

[ March 2016, Score 5 ]

OR

Write a C++ program to accept an integer number and print its reverse

( Hint : If 234 is given, the output must be 432)

[ March 2016, Score 5 ]

**Ans.** #include <iostream>

using namespace std;

```
int main( ) {
int num, temp, rem, res = 0 ;
cout << "Enter a three-digit integer: " ;
cin >> num ;
temp = num ;
```

```
while ( temp != 0) {
    rem = temp % 10 ;
    result += rem * rem * rem ;
    temp /= 10 ;
```

```

}
if (res == num)
    cout << num << " is an Armstrong number.";
else
    cout << num << " is not an Armstrong number.";
return 0;
}

```

OR

```

#include <iostream>
using namespace std ;

int main()
{
    int n, rev = 0, rem;

    cout << "Enter an integer: " ;
    cin >> n ;
    while(n != 0) {
        rem = n%10;
        rev = rev*10 + rem ;
        n /= 10;
    }
    cout << "Reversed Number = " << rev;
    return 0;
}

```

28. Which selection statement tests the value of a variable or an expression against a list of integers or character constants?

a) for            b) if            c) switch            d) conditional expression [July 2015, Score 1 ]

**Ans.** switch

29. .... is an entry controlled loop.

**Ans.** for or while

30. Write a C++ program to display the roots of quadratic equation.

OR



Write a C++ program to display Fibonacci series.

[ July 2015, Score 5 ]

**Ans.**

```
#include <iostream>
#include <cmath>
using namespace std ;
int main() {
    float a, b, c, x1, x2, d, real, imag ;
    cout << "Enter coefficients a, b and c: ";
    cin >> a >> b >> c ;
    d = b * b - 4 * a * c ;
    if (d > 0) {
        x1 = ( - b + sqrt ( d ) ) / ( 2 * a ) ;
        x2 = ( - b - sqrt ( d ) ) / ( 2 * a ) ;
        cout << "Roots are real and different." << "\n" ;
        cout << "x1 = " << x1 << endl ;
        cout << "x2 = " << x2 << endl ;
    }
    else if (d == 0) {
        cout << "Roots are real and same." << "\n" ;
        x1 = - b / ( 2 * a ) ;
        cout << "x1 = x2 = " << x1 << "\n";
    }
    else {
        real = - b / ( 2 * a ) ;
        imag =sqrt(-d)/(2*a);
        cout << "Roots are complex and different." << "\n" ;
        cout << "x1 = " << real << "+" << imag << "i" << "\n" ;
        cout << "x2 = " << real << "-" << imag << "i" << "\n" ;
    }
    return 0;
}
```

**OR**

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

```

int main()
{
    int n, t1 = 0, t2 = 1, next = 0 ;
    cout << "Enter the number of terms: " ;
    cin >> n ;
    cout << "Fibonacci Series: ";
    for (int i = 1; i <= n; ++i)
    {
        // Prints the first two terms.
        if(i == 1)
        {
            cout << " " << t1 ;
            continue;
        }
        if(i == 2)
        {
            cout << t2 << " ";
            continue;
        }
        next = t1 + t2 ;
        t1 = t2;
        t2 = next;
        cout << next << " " ;
    }
    return 0;
}

```

31. a) ----- statement takes the program control out of loop even though the test expression is true

[ March 2015,Score 1 ]

b) Consider the following code fragment. How many times will the character '\*' be printed on the screen?

```

for ( i = 0 ; i < 10 ; i = i + 2 ) ;

```

```

{
    cout<<" * " ;
}

```

[ March 2015, Score 1 ]

**Ans.** a) break

b) One

**32.** Write a program to do the following :

- a) inputs the values for variables n and m.
- b) prints the numbers between 1 and n which are exactly divisible by m.
- c) checks whether the numbers divisible by m are odd or even.

**OR**

Write a program using nested loop that inputs a number n which generates an output as follows.

Hint : if the value of n is 5, the output will be as

```
25
25  16
25  16  9
25  16  9  4
25  16  9  4  1
```

**Ans.**

a) main()

```
{
int m,n;
cout<<"Enter m and n";
cin>>m>>n;
```

b) for ( i =1; i < n ; i++ )

```
{
if ( i % m ==0 )
cout<<i ;
}
```

c)

```
for( i = 1 ; i < n ; i++ )
{
if ( i % m ==0 )
{
if ( i % 2 ==0 )
cout<<" even " ;
else
cout<<" Odd " ;
}
}
```

OR

```
#include<iostream>
using namespace std;
int main()
{
    int i,j;
    for ( i = 5 ; i >= 1 ; i-- )
        {
            for ( j = 5 ; j >= 1 ; j-- )
                cout<< j * j <<" \t " ;
            cout<<" \n " ;
        }
}
```

\*\*\*\*\*

## Chapter 8 – Arrays

1. Compare Linear search and Binary search.

[ March 2020, Score 2 ]

Ans.

Linear search method	Binary search method
The elements need not be in any order	The elements should be in sorted order
Takes more time for the process	Takes very less time for the process
May need to visit all the elements	All the elements are never visited
Suitable when the array is small	Suitable when the array is large