

NAME: HIDAYAT UR RAHMAN

ID: 15125

BS (SE)

SUB: PROGRAMMING FUNDAMENTALS

SIR: DR. FAZAL-E-MALIK

Hidayat-ur-Rahman
id: 15125

(1)

Q No 1: a):-

else..if Statement

Ans:- The else..if statement is used when you need to check multiple conditions within the program. Nesting of if-else blocks can be avoided using else..if statement.

Syntax of else..if Statement

```
if (condition1)
{
    // These statements would execute if the
    // condition1 is true
}
else..if (condition2)
{
    // These statements would execute
    // if the condition2 is true
}
else..if (condition3)
{
    // These statements will execute if the
    // condition3 is true
}
else
{
    // These statements will execute if all the
    // conditions return false
}
```

(2)

Q1 b:-

Ans

```
#include <iostream>
```

```
main()
```

```
{
```

```
int a = 10;
```

```
int b = 6;
```

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

```
{
```

```
cout << "a is Largest number";
```

```
}
```

```
else
```

```
{
```

```
cout << "Smallest";
```

```
}
```

```
}
```

Q2 a:-

Logical operators:-

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

&& (Logical AND):-

This operator is used to combine two conditions.
 — true if both conditions are true
 if (gender == 1 && age >= 65)
 Senior ++

|| (Logical OR):-

— true if either of condition is true
 if (Semester Avg >= 90 || final Exam >= 90)
 cout << ("Student grade is A");

! (Logical NOT):-

This operator takes true to false and vice versa. it is typically used with Boolean. when used with non-Boolean values, it returns false if its single operand can be converted to true; otherwise return true.

`m1 = !true // it returns false`

`m2 = !false // it returns true`

`m3 = !'' // it returns true`

`m4 = !'cat' // it returns false`

x — x — x — x — x — x — x

(5)

Q2B:-

```
#include <iostream>
main ()
{
    float fahr, cel;
    char options;
    cout << "Choose from following options:" << endl;
    cout << "1 celcius to fahrenheit;" << endl;
    cout << "2 fahrenheit to celcius;" << endl;
    cin >> options;
    // options for converting fahrenheit to celcius to fahrenheit
    if (option == 1)
    {
        cout << "enter the temperature in celcius:";
        cin >> celcius;
        fahrenheit = (1.8 * celcius) + 32.8;
        // temp conversions formula
        cout << "In temp in degree fahrenheit:" << fahrenheit
        << "f" << endl;
    }
    // option for converting fahrenheit into celcius
    elseif (options == 2)
```

(b)

```
{  
    cout << "enter the temperature in faranheit:";  
    cin >> faranheight;  
    celcius = (faranheight - 32) / 1.8;  
    // temp conversion formula  
    cout << "\n temperature in degree celcius:  
            " << celcius << "c" << endl;  
}  
else  
    cout << "Error wrong input." << endl;  
return 0;  
}
```

Q3a:-

LOOP

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

In C++ we have three types of basic loops.

- ① for loop
- ② while loop
- ③ do while loop

① for Loop:-

This loop is used to execute a set of statements repeatedly until a particular condition is satisfied. In for loop exactly two semicolons, one after initialization and second after conditions.

In this loop we can have more than one initialization or increment/decrement.

(8)

Separated using comma operators.
for loop can have only one condition.

Syntax for for loop:-

```
for (initialization condition; termination  
condition; increment condition)  
{  
    Statement (S);  
}
```

(2) While Loop:-

This can be address
as an entry control loop. its completed
in 3 steps.

- ① variable initialization. (e.g int x = 0;)
- condition (e.g while (x <= 10))
- variable increment or decreament (x++
or x-- or x = x + 2)

↳ While loop execute zero or more
time. what if we want the loop
to execute at least on time.

Syntax for while loop:-

```
while (condition)  
{  
    Statement; variable increment or  
decrement; }.
```


(3) do-while Loop:-

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

↳ Do-while execute one or more time.

Syntax of do-while Loop:-

```
do { // a couple of statements }
```

```
  }
  while (condition);
```

~~Ans:-~~ x — x — x — x — x — x — x

Q36:-

```
#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;
}
```

x — x — x — x — x — x

QNO4

(a)

Break:

break is used to break or terminate a loop whenever we want just type break; after the statement after which you want to break the loop as simple as that! Remember we had also used break statements in switch.

Consider an example

```
#include <iostream>
int main () {
    using namespace std;
    int n;
    for (n=1; n<=5; n++) {
        cout << "*" << endl;
        if (n==2) {
            break;
        }
    }
}
```

Continue Statement:-

Continue Statement works similar to break statement. The only difference is that break statement terminates the loop

(12)

where continue statement passes control to the conditional test i.e, where the conditional is checked

in short it passes control on the nearest conditional test in do while loop, or the condition of while and while loop, or the condition of for in statements skipping the rest of the statement in the loop

```
include <iostream>
```

```
int main () {
```

```
using namespace std;
```

```
int n=1;
```

```
while (n < 10) {
```

```
    n = n + 1;
```

```
    continue;
```

```
}
```

```
cout << "n = " << n << endl;
```

```
}
```

```
return 0;
```

```
}
```

```
x — x — x — x — x
```

Q4b:-

```
#include <iostream>
#include <conio.h>
main ()
{
    int sum;
    cout << "The first 10 natural number are:
        \n";
    for (i = 1; i <= 10; i++)
    {
        cout << i << " ";
        sum = sum + i;
    }
    cout << "\n The sum of first 10
        natural number is \n";
    cout << sum;
    return 0;
}
```

x ——— x ——— x ——— x ——— x ——— x

Q5
Ans

Character Set:

In C++ Character set is a set of all valid characters that can be used in C++ programme. Character set is used to specify characters or symbols recognized by language. Character set is a set of all valid characters that can be used to form words, numbers and expressions in source programs. The source character set consists of the characters used for the source characters program text, while the execution character set is the set of characters used for the source program text while the execution character set is the set of characters used during the execution of the programme. It is not necessary that source character set match and execution character set are same. Includes following

Characters.

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

the 26 uppercase Roman characters
A, B, C, D, E, F, G, H, I, J, K, L, M,
N, O, P, Q, R, S, T

Constants:-

Constants refer to immutable values. Constants are basically literals whose values can not be modified or changed during the execution of program. Constants are also called as literal. Constants must be initialized when declared as values can not be assigned it to later. In C++ constants can be following four basic types

variable:-

variable is an identified used to refer memory location in computer memory that holds a value for that

(16)

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 for that variable. The size of memory block allocated in type of the value it holds is completely dependent upon the type of variable.

Key words:-

in C++ there are as a set of reserved words that you can not use is identified. These words are known as reserved words or key words. Key words are standard identifiers and their functions is pre defined by the compiler. We can not use keywords as variable names e.g. classname or method names or as any other identifiers.

Relational operator:

A relational operator is used to check the relationship between two operands. eg
// checks if a is greater than b

$a > b;$

Here ~~for~~ $>$ is a relational operator. it checks if a is greater than b or not. if the relation is true, it returns 1 whereas if the relation is false it returns 0.

x — x — x — x — x

End