NAME JIBRAN
ID 14472
SUBJECT PROGRAMMING FUNDAMENTAL
SEMESTER 4th
DATE 29/09/2020

Q (1) part (a)

Purpose of if statement:-

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

Forms of if statement:-

Two types of if conditions are IF-THEN and IF-THEN-ELSE Conditions.

IF Condition.

```
#include <iostream
main()
 {
  if(condition)
  {
   //Body of if statement
  }
}
```

Jibran
14472

// IF-THEN-ELSE condition:-

```
main()
{
  if(condition)
  {
  }
  else{
  }
}
```

Q (1) part (b)

```
#include<iostream>
#include<conio.h>
using namespace std;

int main()
{
  int n1, n2;
  Cout<<"Enter two numbers 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;
}
```

Jibran
14472

Day. M T W T F S

Date: __/__/__

Q(2) part (a)

Logical Operators:-

Logical operators are used with binary variables. They are mainly used in conditional statements.

Logical operator in C++ are
&&, ||, !

(1) && (logical AND):

It is used to combine two condition.

- True if both conditions are true

```
if (gender == 1 && age >= 74)
```

(2) || (logical OR):

- True if either of a condition is true.

```
(a == b) || (c < b)
```

In this example only one condition is true so its answer will also be true.

(3) ! (logical NOT):

- Returns true when its condition is false, & vice versa.

Jibran
14472

Day: M T W T F S

Date: __/__/__

- True only if operand is 0.

if c=5 then expression
!(c==5)
equals to zero.

Q(2) part (b)

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

Checked By:..... Parents:..... Excellent ☐ Good ☐

Q (3) part (a)

What does looping mean? ---->

* 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:
- for loop
- while loop
- do-while loop

* for loop:-
A for loop is a more efficient loop structure in 'C' programming. The general structure of for loop is as follows:

```
for(inital value; condition; incrementation
                              or decrementation)
{
  statements;
}
```

(1) The initial value of the for loop is performed only once.
(2) The condition is a Boolean expression that tests and compares the counter to a fixed value after each iteration, stopping the for loop when false is returned.

Jibran
14472

Day: M T W T F S Date: __/__/__

(3) The incrementation/decrementation increases (or decreases) the counter by a set value.

* **While loop:-**

The basic format of while loop is

```
while (condition) {
        statements;
}
```

It is an entry control loop. In while loop, a condition is evaluated before processing a body of the loop. If a condition is true then the body of a loop is executed. After the body of a loop is executed then control again goes back at the beginning and the condition is checked if it is true, the same process is executed until the condition becomes false. Once the condition becomes false, the control goes out of the loop.

* **do-while loop:**

A do-while loop is similar to the while loop except that the condition is always executed after the body of a loop. It is also called an exit-controlled loop.

Jibran
14472

(7)

Day. M T W T F S

Date: __/__/__

In the do-while loop, the body of a loop is always executed at least once. After the body is executed, then it checks the condition. If the condition is true, then it will again execute the body of a loop otherwise control is transferred out of the loop.

Checked By:..... Parents:.....

Day: M T W T F S

Date: __/__/__

Jibran
14472

Q(3) part (b)

(b) Write a C++ to read . . . . . Even or Odd number?

```
#include<iostream>
#include<conio.h>
using namespace std;

int 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;
}
```

Checked By:..... Parents:..... Excellent ☐ Good ☐ BABAR PAPER PRODUCTS

Jibran
14472

Day. M T W T F S

Date: __/__/__

Q(4) part (a)

** The break Statement:-

(1) There are situations where we want to jump out of a loop instantly, waiting to get back to the conditional test.

(2) The keyword break allows us to do this.

(3) When break is encountered inside any loop, control automatically passes to the first statement after the loop.

(4) A break is usually associated with an if.

(5) The keyword break, breaks the control only from the loop in which it is placed.

** Continue Statement:-

(1) Continue statement allows to take the control to the beginning of the loop, by passing the statements inside the loop, which have not yet been executed.

(2) A continue is usually associated with an if.

Jibran
14472

Day. M T W T F S

Date: __/__/__

Q (4) part (b)

```
#include<iostream>
#include<conio.h>
using namespace std;
int main()
{
    int i;
    int sum = 0;

    Cout<<"The First 10 natural numbers are:\n";
    for(i = 1; i<=10; i++)
    {
        Cout<<i<<" ";
        sum = sum + i;
    }
    Cout<<"\nThe sum of First 10 natural number is\n";
    Cout<<sum;
    return 0;
}
```

Q(5)

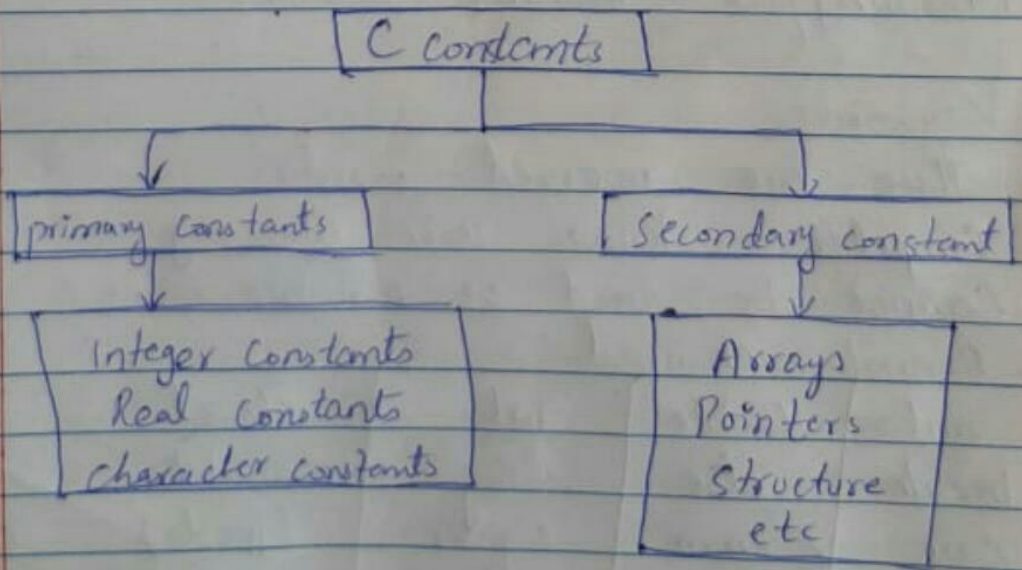
(a) C++ character set:

(1) Alphabets
A, B, ... Y, Z

(2) Digits
0, 1, 2, 3, 4, 5 ..... 8, 9

(3) Special symbols
~ ! # $ % ^ & * ( ) _ - = + ...

(b) Constants:-
Constant is an entity that doesn't change.



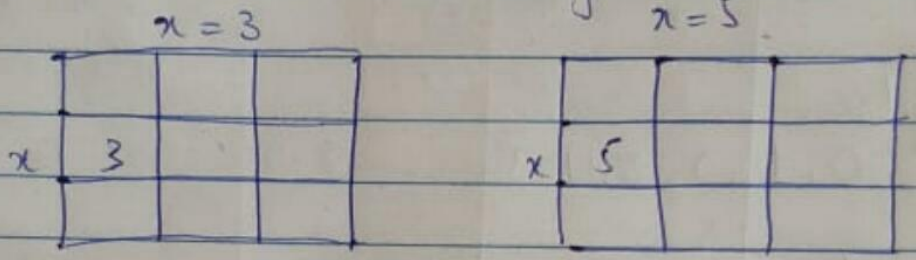
Jibran
14472

Day. M T W T F S Date: __/__/__

(c) Variables:-

An entity that may change during program execution.

(2) These are names given to locations in memory.



(3) Series of characters (letters, digits, underscores)

(4) Must begin with a letter or underscore.

(5) Case sensitive.

(6) Meaningful naming scheme.

(d) Keywords:

(1) These are reserved words.

(2) Compiler knows their meaning

(3) Cannot be used as variable name.

(4) Cannot changed.

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	sizeof	volatile

Jibran
14472

Day. M T W T F S

Date: __/__/__

do	if	static	while

(c) Relational Operators:

Standard algebric	C++ equality	Example	Meaning
$>$	$>$	$x>y$	x is greater than y
$<$	$<$	$x<y$	x is less than y
$\geq$	$>=$	$x>=y$	x is greater or equal to y
$\leq$	$<=$	$x<=y$	x is less than or equal to y
Equality operators.			
$=$	$==$	$x==y$	x is equal to y
$\neq$	$!=$	$x!=y$	x is not equal to y