***Final-Term ASSIGNMENT***

***NAME :- Muhammad Daniyal***

***ID :- 17011***

***Program :-BS(CS)***

***Subject :- Programming fundamental***

***Lecturer :- Dr. Fazal-E-Malik***

**Note: Attempt all Questions. Help can be taken from net where ever is required.**

**Question No 1**

**Part A**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  | What is the purpose of ***if statement***? Discuss its two different forms with  Examples.  **ANSWER:**  IF statement is a programming conditional statement which we used to check a condition and if the condition is true, we run a block of statements and if the IF condition is false the code outside of the IF statement will be executed  **Forms of IF statement:**  The two different forms of IF statement given below   1. IF Statement   Using IF statement we apply 1 condition at a time  Example:  if (20 > 18) {   cout << "20 is greater than 18";  }   1. Else If Statement   Using else if statement we used multiple conditions at a time  int time = 22;  if (time < 10) {   cout << "Good morning.";  }  else if (time < 20) {   cout << "Good day.";  }  else {   cout << "Good evening.";  }  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* | **5** |

**Question No 1**

**Part B**

Write a C++ program to read two numbers from keyboard and then find the   
LARGEST number of them.

**ANSWER:**

#include<iostream>

using namespace std;

int main(){

int number1,number2;

cout<<"enter a number1 :";

cin>>number1;

cout<<"enter a number2 :";

cin>>number2;

if(number1>number2)

cout<<number1 <<" is greater then "<<number2;

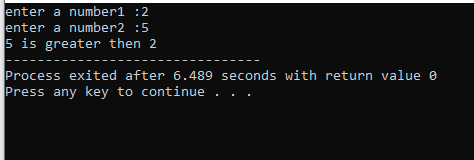
else

cout<<number2 <<" is greater then "<<number1;

return 0;

}

Output:



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Question No 2**  **Part A**  What are the Logical Operators? Explain them  **ANSWER:**  **Logical Operators:**  In C++ language we use Logical operators to allow a program to make a decision based on multiple conditions. Each operand is considered a condition that can be evaluated to a true or false value.  Some of the Logical operators used in C++ Language are given below,   |  |  | | --- | --- | | **Operator** | **Description** | | && | This operator is called Logical AND operator. If both the operands are non-zero, then condition becomes true. | | || | This operator is called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true. | | ! | This operator is called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true, then Logical NOT operator will make false. |   \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* |

**Question No 2**

**Part B**

Write a C++ program to get Temperature in Fahrenheit ***F*** and then find the Atmosphere according to the below rules:

* If temperature ***F*** is above 40 degree Fahrenheit then display…………………..Very Hot.
* If temperature ***F***is between 35 & 40 degree Fahrenheit then display………Tolerable.
* If temperature ***F*** is between 30 & 35 degree Fahrenheit then display………Warm.
* If temperature ***F*** is less than 30degree Fahrenheit then display………….……Cool.

**ANSWER:**

#include<iostream>

using namespace std;

int main(){

int t,f;

cout<<"enter temp celsius:";

cin>>t;

cout<<endl;

f=(9/5\*t)+32;

cout<<f;

cout<<endl;

if(f>40)

cout<<"degree fehrenhent is very hottt :";

else if(f==35 && f==40)

cout<<"degree fehrenhent is tolerable :";

else if(f==30 && f==35)

cout<<"degree fehrenhent is warm :";

else if(f<30)

cout<<"";

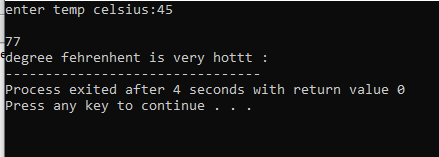
else

cout<<"degree fehrenhent is cool:";

return 0;

}

Output



**Question No 3**

**Part A**

What does ***Looping*** mean? Explain different loops in C++.

**ANSWER:**

In C++ language we use loops to execute a statement or group of statements multiple times.

Different types of loops in C++ language are as follow below,

**While Loop:**

A While loops is a loop in which any statement repeatedly executes a target statement as long as a given condition is true.

Syntax:

while(condition) {

statement(s);

}

Example:

#include <iostream>

int main () {

int a = 10;

while( a < 20 ) {

cout << "value of a: " << a << endl;

a++;

}

return 0;

}

**For Loop:**

A for loop is a repetition control structure that allows us to efficiently write a loop that needs to execute a specific number of times.

Syntax:

for ( init; condition; increment ) {

statement(s);

}

Example:

#include <iostream>

using namespace std;

int main () {

for( int a = 10; a < 20; a = a + 1 ) {

cout << "value of a: " << a << endl;

}

return 0;

}

**Do-While Loop:**

Unlike for and while loops, which test the loop condition at the top of the loop, the do...while loop checks its condition at the bottom of the loop.

A do...while loop is similar to a while loop, except that a do...while loop is guaranteed to execute at least one time.

Syntax:

do {

statement(s);

}

while( condition );

Example:

[Live Demo](http://tpcg.io/RKwBEo)

#include <iostream>

using namespace std;

int main () {

int a = 10;

do {

cout << "value of a: " << a << endl;

a = a + 1;

} while( a < 20 );

return 0;

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question No 3**

**Part B**

Write a C++ program to read a number from keyboard and then determine whether it is ***Even or Odd*** number?

**ANSWER:**

#include<iostream>

using namespace std;

int main(){

int number;

cout<<"enter number for a checking :";

cin>>number;

if(number%2==0)

cout<<"number is even :" <<number;

else

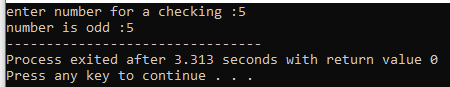
cout<<"number is odd :"<<number;

return 0;

}

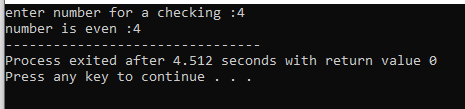
**Output 1:**

**Checking For Odd:**



Output 2:

Checking for Even :



**Question NO 4**

**Part A**

What is the purpose of using ***break and continue statements***?

**ANSWER:**

**Break Statement:**

The break statement can be used to jump out of a loop or switch statement.

Example:

for (int i = 0; i < 10; i++) {  
  if (i == 4) {  
    break;  
  }  
  cout << i << "\n";  
}

**Continue Statement:**

The continue statement breaks one iteration in the loop if a specified condition occurs, and continues with the next iteration in the loop.

Example:

for (int i = 0; i < 10; i++) {  
  if (i == 4) {  
    continue;  
  }  
  cout << i << "\n";  
}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Question No 4**

**Part B**

Write a C++ program to find the sum of the following numbers:

**1+2+3+………+10**

**ANSWER:**

#include<iostream>

using namespace std;

int main(){

int number,sum=0;

cout<<"enter a number for summing :";

cin>>number;

int i=0;

while(i<=number){

sum=sum+i;

i++;

cout<<sum;

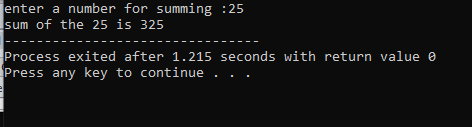
}

cout<<"sum of the "<<number<<" is "<<sum;

return 0;

}

Output:



|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **Question No 5**  What is an array? Explain One-Dimensional and Two-Dimensional Arrays with examples.  **ANSWER:**  **Array:**  An array is a series of elements of the same type placed in contiguous memory locations that can be individually referenced by adding an index to a unique identifier.  for example, we have five values of type int and it can be declared as an array without having to declare 5 different variables each with its own identifier. Instead, using an array, the five int values are stored in contiguous memory locations, and all five can be accessed using the same identifier, with the proper index.  **One Dimensional Array:**  A one-dimensional array is a group of elements having the same datatype and same name. Individual elements are referred to using common name and unique index of the elements.  The simplest form of an array is one-dimensional-array. The array itself is given name and its elements are referred to by their subscripts.  **Declare One Dimensional Array in C++:**  Here is the general form to declare one dimensional array in C++  data\_type array\_name[array\_size];  Here, data\_type is any valid C++ data type, array\_name is the name of the array, and array\_size is the size of array. Here is an example, declaring an array named arr of int type, having maximum element size of 10 elements  int arr[10];  Here is an example, declaring and initializing values to the array name arr of type int, containing 10 elements  int arr[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};  **Applying one Dimensional Array in C++:**  #include<iostream.h>  #include<conio.h>  void main()  {  clrscr();  int arr[5] = {1, 2, 3, 4, 5};  int i;  for(i=0; i<5; i++)  {  cout<<"arr["<<i<<"] = "<<arr[i]<<"\n";  }  getch();  }  **Two Dimensional Array:**  A two-dimensional array can be think as a table, which will have x number of rows and y number of columns. For example we have a 2-dimensional array arr, which contains three rows and four columns can be shown as below:  **Initialization:**  int a[3][4] = { {0, 1, 2, 3} , {4, 5, 6, 7} , {8, 9, 10, 11} };  The nested braces, which indicate the intended row, are optional. The following initialization is equivalent to previous example  int a[3][4] = {0,1,2,3,4,5,6,7,8,9,10,11};  **Applying Two Dimensional Array in C++:**  #include <iostream>  using namespace std;    int main () {  int a[5][2] = { {0,0}, {1,2}, {2,4}, {3,6},{4,8}};    for ( int i = 0; i < 5; i++ )  for ( int j = 0; j < 2; j++ ) {    cout << "a[" << i << "][" << j << "]: ";  cout << a[i][j]<< endl;    }  return 0;  } | **10** |
|  |  |  |  |