ASSIGNMENT-1

1. **A/ WRITE A C++ PROGRAM TO PRINT YOUR NAME AND CITY.**

**PROGRAM**

*#include<iostream>*

*using namespace std;*

*int main() {*

*cout << "Name :: Abhisek Roy" << endl*

*<< "Address :: Salua colony near Red Cross Hospital,India" << endl;*

*}*

**OUTPUT**

*Name :: Abhisek Roy*

*Address :: Salua colony near Red Cross Hospital,India*

**B/ WHAT DO MEAN BY SINGLE LINE AND MULTI LINE COMMENT ?**

Single Line Comments are useful where you need to comment or hide two-three words however multiple line comments are useful where you want to hide multiple lines of code.

/\* This is Multiline Comment \*/

//single line comment

1. **WRITE PROGRAM WHETHER THE AGE LIMET OF THE EMLOYEE IS 20 TO 60 USING SWITCH CASE STATEMENT.**

#include <iostream>

using namespace std;

int main() {

int age=18;

cout<<"Enter your age = ";

cin>>age;

switch(age) {

case'18-30':

cout<<"salary is 20000 "<<age;

break;

case'31-50':

cout<<"salary is 45000"<<age;

break;

case'51-60':

cout<<"salary is 60000 "<<age;

break;

default:

cout<<"invalid age entered";

break;

}

return 0;

}

**OUTPUT**

Enter your age=20

Salary is 20000

ASSIGNMENT-2

**WRITE PROGRAM TO DISPLAY TWO NUMBERS FORM THE USER.**

#include <stdio.h>

int main()

{

int a,b;

printf("Enter two number\n");

scanf("%d",&a);

scanf("%d",&b);

printf("The number is %d ,%d",a,b);

return 0;

}

**OUTPUT**

Enter two number

21

26

The number is 21,26

B/ **WRITE A PROGRAM TO CHECK WHETHER THE NUMBERS ARE EQUAL OR NOT.**

#include <stdio.h>

void main()

{

int int1, int2;

printf("Input the values for Number1 and Number2 : ");

scanf("%d %d", &int1, &int2);

if (int1 == int2)

printf("Number1 and Number2 are equal\n");

else

printf("Number1 and Number2 are not equal\n");

}

**OUTPUT**

Input the values for Number1 and Number2 :21

21

Number1 and Number2 are equal

**C++ PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int num1,num2;

cout << "\n\n Compare the first number with second number numbers:\n";

cout << "---------------------------------------------------------\n";

cout << " Input the first integer: ";

cin>> num1;

cout << " Input the second integer: ";

cin>> num2;

if ( num1 == num2 )

cout<< num1<< " IS EEQUAL " << num2 <<endl;

if ( num1 != num2 )

cout<< num1<<" IS NOT EQUAL " << num2<<endl;

}

**OUTPUT**

Input the first integer:21

Input the first integer:21

21 IS EQUAL 21

3.**WRITE A PROGRAM TO DISPLAY THE 10 NATURAL NUMBER.**

**PROGRAM**

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

#include <stdio.h>

void main()

{

int i;

printf("The first 10 natural numbers are:\n");

for (i=1;i<=10;i++)

{

printf("%d ",i);

}

printf("\n");

}

**OUTPUT**

The first 10 natural numbers are:

1 2 3 4 5 6 7 8 9 10

**C++ PROGRAM**

#include <iostream>

using namespace std;

int main()

{

int i;

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

cout << "---------------------------------------\n";

cout << " The natural numbers are: \n";

for (i = 1; i <= 10; i++)

{

cout << i << " ";

}

cout << endl;

}

**OUTPUT**

Find the first 10 natural numbers

1 2 3 4 5 6 7 8 9 10

**ASSIGNMENT-3**

1. **WRITE A C PROGRAM TO TAKE 2D ARRAY AS INPUT FROM USER**

#include<stdio.h>

int main(){

/\* 2D array declaration\*/

int disp[2][3];

/\*Counter variables for the loop\*/

int i, j;

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

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

printf("Enter value for disp[%d][%d]:", i, j);

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

}

}

//Displaying array elements

printf("Two Dimensional array elements:\n");

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

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

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

if(j==2){

printf("\n");

}

}

}

return 0;

}

**OUTPUT**

Enter value for disp[0][0]:1

Enter value for disp[0][1]:2

Enter value for disp[0][2]:6

Enter value for disp[1][0]:7

Enter value for disp[1][1]:8

Enter value for disp[1][2]:9

Two Dimensional array elements:

1 2 6

7 8 9

1. **B/ WRITE A C PROGRAM TO TAKE 2D ARRAY AS INPUT FROM USER AND SUBSTRACT THEM THE RESULT.**

#include <stdio.h>

int main()

{

int rowCount, columnCount, i, j;

int firstMatrix[10][10], secondMatrix[10][10], resultMatrix[10][10];

printf("Number of rows of matrices to be subtracted : ");

scanf("%d", &rowCount);

printf("Number of columns matrices to be subtracted : ");

scanf("%d", &columnCount);

printf("Elements of first matrix : \n");

for (i = 0; i < rowCount; i++)

for (j = 0; j < columnCount; j++)

scanf("%d", &firstMatrix[i][j]);

printf("Elements of second matrix : \n");

for (i = 0; i < rowCount; i++)

for (j = 0; j < columnCount; j++)

scanf("%d", &secondMatrix[i][j]);

printf("Difference of entered matrices : \n");

for (i = 0; i < rowCount; i++)

{

for (j = 0; j < columnCount; j++)

{

resultMatrix[i][j] = firstMatrix[i][j] - secondMatrix[i][j];

printf("%d\t",resultMatrix[i][j]);

}

printf("\n");

}

return 0;

}

**OUTPUT**

Number of rows of matrices to be subtracted : 3

Number of columns matrices to be subtracted : 3

Elements of first matrix :

6

6

6

6

6

6

6

6

6

Elements of second matrix :

3

3

3

3

3

3

3

3

3

Difference of entered matrices :

3 3 3

3 3 3

3 3 3

**WRITE A C++ PROGRAM TO TAKE 2D ARRAY AS INPUT FROM USER AND SUBTRACT THEM AND FIND THE RESULT.**

#include<iostream>

using namespace std;

int main()

{

int row, col, m1[10][10], m2[10][10], sum[10][10];

cout<<"Enter the number of rows(should be >1 and <10): ";

cin>>row;

cout<<"Enter the number of column(should be >1 and <10): ";

cin>>col;

cout << "Enter the elements of first 1st matrix: ";

for (int i = 0;i<row;i++ ) {

for (int j = 0;j < col;j++ ) {

cin>>m1[i][j];

}

}

cout << "Enter the elements of first 1st matrix: ";

for (int i = 0;i<row;i++ ) {

for (int j = 0;j<col;j++ ) {

cin>>m2[i][j];

}

}

cout<<"Output: ";

for (int i = 0;i<row;i++ ) {

for (int j = 0;j<col;j++ ) {

sum[i][j]=m1[i][j]+m2[i][j];

cout<<sum[i][j]<<" ";

}

}

return 0;

}

**OUTPUT**

Number of rows of matrices to be subtracted : 3

Number of columns matrices to be subtracted : 3

Elements of first matrix :

6

6

6

6

6

6

6

6

6

Elements of second matrix :

3

3

3

3

3

3

3

3

3

Difference of entered matrices :

3 3 3

3 3 3

3 3 3

1. **write a c program to concatenate two string BEST AND FRIENDS STRING**

#include <stdio.h>

int main() {

char s1[100] = "BEST ", s2[] = "FRIENDS";

int length, j;

// store length of s1 in the length variable

length = 0;

while (s1[length] != '\0') {

++length;

}

// concatenate s2 to s1

for (j = 0; s2[j] != '\0'; ++j, ++length) {

s1[length] = s2[j];

}

// terminating the s1 string

s1[length] = '\0';

printf("After concatenation: ");

puts(s1);

return 0;

}

**OUTPUT**

After concatenation: BEST FRIENDS

**C++ PROGRAM**

#include <iostream>

using namespace std;

int main()

{

string s1, s2, result;

cout << "Enter string s1: ";

getline (cin, s1);

cout << "Enter string s2: ";

getline (cin, s2);

result = s1 + s2;

cout << "Resultant String = "<< result;

return 0;

}

OUTPUT

Enter string s1: BEST

Enter string s2: FRIENDS

Resultant String =BEST FRIENDS

1. **Are strncpy and strcpy same? Justify your answer.**

NO

strcpy( ) function copies whole content of one string into another string. Whereas, strncpy( ) function copies portion of contents of one string into another string. If destination string length is less than source string, entire/specified source string value won't be copied into destination string in both cases.

**ASSIGNMENT-4**

**1.** **Write a program to display records like name, address, designation and salary of an**

**employee.**

#include <stdio.h>

/\*structure declaration\*/

struct employee{

char name[30];

int empId;

float salary;

};

int main()

{

/\*declare structure variable\*/

struct employee emp;

/\*read employee details\*/

printf("\nEnter details :\n");

printf("Name ?:");

gets(emp.name);

printf("ID ?:");

scanf("%d",&emp.empId);

printf("Salary ?:");

scanf("%f",&emp.salary);

/\*print employee details\*/

printf("\nEntered detail is:");

printf("Name: %s" ,emp.name);

printf("Id: %d" ,emp.empId);

printf("Salary: %f\n",emp.salary);

return 0;

}

**OUTPUT**

Enter details :

Name ?:Abhisek Roy

ID ?:1245

Salary ?:50000

Entered detail is:Name: Abhisek RoyId: 124Salary: 50000.000000

**C++ PROGRAM**

#include <iostream>

using namespace std;

struct employee {

int empID;

char name[50];

int salary;

char department[50];

};

int main() {

struct employee emp[3] = { { 1 , "Harry" , 20000 , "Finance" } , { 2 , "Sally" , 50000 , "HR" } , { 3 , "John" , 15000 , "Technical" } };

cout<<"The employee information is given as follows:"<<endl;

cout<<endl;

for(int i=0; i<3;i++) {

cout<<"Employee ID: "<<emp[i].empID<<endl;

cout<<"Name: "<<emp[i].name<<endl;

cout<<"Salary: "<<emp[i].salary<<endl;

cout<<"Department: "<<emp[i].department<<endl;

cout<<endl;

}

return 0;

}

OUTPUT

The employee information is given as follows:

Employee ID: 1

Name: Harry

Salary: 20000

Department: Finance

Employee ID: 2

Name: Sally

Salary: 50000

Department: HR

Employee ID: 3

Name: John

Salary: 15000

Department: Technical

**2.**

**a) Write a C and C++ program, declare a variable and a pointer.**

**b) Store the address of the variable in the pointer.**

**c) Print the value of pointer.**

**A.**

datatype \*var\_name;

int \*ptr; //ptr can point to an address which holds int data

* Define a pointer variable
* Assigning the address of a variable to a pointer using unary operator (&) which returns the address of that variable.
* Accessing the value stored in the address using unary operator (\*) which returns the value of the variable located at the address specified by its operand.

A pointer is a variable that holds a memory address where a value lives. A pointer is declared using the \* operator before an identifier. As C++ is a statically typed language, the type is required to declare a pointer. This is the type of data that will live at the memory address it points to.

B.

In c++ you can get the memory address of a variable by using the & operator, like: cout << &i << endl; The output of that cout is the memory address of the first byte of the variable i we just created

C.

1. Define a pointer variable.
2. Assigning the address of a variable to a pointer using unary operator (&) which returns the address of that variable.
3. Accessing the value stored in the address using unary operator (\*) which returns the value of the variable located at the address specified by its operand.

**3.DIFFERENCE BETWEEN CALL BY VALUE AND CALL BY REFERENCE.**

method original value is not modified whereas, in Call by reference method, the original value is modified.

In Call by value, a copy of the variable is passed whereas in Call by reference, a variable itself is passed.

In Call by value, actual and formal arguments will be created in different memory locations whereas in Call by reference, actual and formal arguments will be created in the same memory location.

Call by value is the default method in programming languages like C++, PHP, Visual Basic NET, and C# whereas Call by reference is supported only Java language.

Call by Value, variables are passed using a straightforward method whereas Call by Reference, pointers are required to store the address of variables.

1. **A/ WRITE A PROGRAM TO CREAT A FILE.**

#include<stdio.h>

int main(){

FILE \*fp;

fp=fopen("file.txt","w");

fclose(fp);

return 0;

}

**C++ PROGRAM**

#include <iostream>

#include <fstream>

using namespace std;

int main() {

// Create and open a text file

ofstream MyFile("filename.txt");

// Write to the file

MyFile << "Files can be tricky, but it is fun enough!";

// Close the file

MyFile.close();

}

B**.WRITE YOUR NAME AND ADDRESS IN YOUR FILE**

#include <stdio.h>

#include <stdlib.h>

int main()

{

FILE \*fptr;

// use appropriate location if you are using MacOS or Linux

fptr = fopen("C:\\program.txt","w");

if(fptr == NULL)

{

printf("Error!");

exit(1);

}

printf("my name is Abhisek Roy \n");

printf("my adress is salua colony near red cross hospital ");

fprintf(fptr,"%d",num);

fclose(fptr);

return 0;

}

**OUTPUT**

my name is Abhisek Roy

my adress is salua colony near red cross hospital

**c. DISPLAY IT ON THE CONSOLE USING C PROGRAM.**

#include <stdio.h>

int main() {

FILE \*fp;

int c;

// open the current input file

fp = fopen(\_\_FILE\_\_,"r");

do {

c = getc(fp); // read character

putchar(c); // display character

}

while(c != EOF); // loop until the end of file is reached

fclose(fp);

return 0;

}

**OUTPUT**

#include <stdio.h>

int main() {

FILE \*fp;

int c;

// open the current input file

fp = fopen(\_\_FILE\_\_,"r");

do {

c = getc(fp); // read character

putchar(c); // display character

}

while(c != EOF); // loop until the end of file is reached

fclose(fp);

return 0;

}