Assignment 1

1

Write a C++ program that inputs a character from keyboard and display its correspondingASCII value on the screen.

Code-

#include <iostream>

using namespace std;

int main() {

char c;

cout << "Enter a character: ";

cin >> c;

cout << "ASCII Value of " << c << " is " << int(c);

return 0;

}

2.

Write a C++ program using class called temperature and memberfunctions for a temperature in Fahrenheit and display it in Celsius.

Code-

#include<iostream>

using namespace std;

// define a class Temperature

class Temperature

{

private:

float fahren, celsius;

public:

float conversion(float f)

{

fahren=f;

celsius=(fahren-32)\* 5.0/9.0;

return celsius;

}

};

int main()

{

// define an object of Temperature class

Temperature t;

float f;

cout<<"Enter Temperature in Fahrenheit=";

cin>>f;

// call conversion function with object t

cout<<"Temperature in Celsius="<<t.conversion(f);

return 0;

}

3.

Write a C++ program function using reference variables as arguments to swap the values of a pair of integers.

Code-

#include<iostream.h>

#include<conio.h>

class abc

{

private:

int t;

public:

void swap(int \*a, int \*b)

{

t=\*a;

\*a=\*b;

\*b=t;

}

};

void main()

{

int a,b;

clrscr();

abc sa;

cout<<"Enter the values of A: ";

cin>>a;

cout<<"Enter the values of B: ";

cin>>b;

sa.swap(&a,&b);

cout<<"\nValues of A is: "<<a;

cout<<"\nValues of B is: "<<b;

getch();

}

4.Write a C++ program which explains and show different storage classes auto, extern staticand register.

Code-

 #include <iostream>

using namespace std;

void autoStorageClass()

{

    cout << "Demonstrating auto class\n";

    // Declaring an auto variable

    // No data-type declaration needed

    auto a = 32;

    auto b = 3.2;

    auto c = "computerlab";

    auto d = 'G';

    // printing the auto variables

    cout << a << " \n";

    cout << b << " \n";

    cout << c << " \n";

    cout << d << " \n";

}

int main()

{

    // To demonstrate auto Storage Class

    autoStorageClass();

    return 0;

}

Extern

#include <iostream>

**using** **namespace** std;

// declaring the variable which is to

// be made extern an initial value can

// also be initialized to x

**int** x;

**void** externStorageClass()

{

    cout << "Demonstrating extern class\n";

    // telling the compiler that the variable

    // x is an extern variable and has been

    // defined elsewhere (above the main

    // function)

**extern** **int** x;

    // printing the extern variables 'x'

    cout << "Value of the variable 'x'"

         << "declared, as extern: " << x << "\n";

    // value of extern variable x modified

    x = 2;

    // printing the modified values of

    // extern variables 'x'

    cout

        << "Modified value of the variable 'x'"

        << " declared as extern: \n"

        << x;

}

**int** main()

{

    // To demonstrate extern Storage Class

    externStorageClass();

**return** 0;

}

Static

#include <iostream>

**using** **namespace** std;

// Function containing static variables

// memory is retained during execution

**int** staticFun()

{

    cout << "For static variables: ";

**static** **int** count = 0;

    count++;

**return** count;

}

// Function containing non-static variables

// memory is destroyed

**int** nonStaticFun()

{

    cout << "For Non-Static variables: ";

**int** count = 0;

    count++;

**return** count;

}

**int** main()

{

    // Calling the static parts

    cout << staticFun() << "\n";

    cout << staticFun() << "\n";

    ;

    // Calling the non-static parts

    cout << nonStaticFun() << "\n";

    ;

    cout << nonStaticFun() << "\n";

    ;

**return** 0;

}

Register

#include <iostream>

**using** **namespace** std;

**void** registerStorageClass()

{

    cout << "Demonstrating register class\n";

    // declaring a register variable

**register** **char** b = 'G';

    // printing the register variable 'b'

    cout << "Value of the variable 'b'"

         << " declared as register: " << b;

}

**int** main()

{

    // To demonstrate register Storage Class

    registerStorageClass();

**return** 0;

}

*Assignment 2*

1.Write a C++ program to find factorial of number using function.

Code-

#include <iostream>

**using** **namespace** std;

**int** main()

{

**int** i,fact=1,number;

  cout<<"Enter any Number: ";

 cin>>number;

**for**(i=1;i<=number;i++){

      fact=fact\*i;

  }

  cout<<"Factorial of " <<number<<" is: "<<fact<<endl;

**return** 0;

}

2. Write a C++ program to check number is palindrome or not using Function.

Code-

#include<iostream>

using namespace std;

int checkPalindrome(int);

int main()

{

int num, val;

cout<<"Enter the Number: ";

cin>>num;

val = checkPalindrome(num);

if(val==0)

cout<<"\nIt is a Palindrome Number";

else

cout<<"\nIt is not a Palindrome Number";

cout<<endl;

return 0;

}

int checkPalindrome(int n)

{

int temp, rem, rev=0;

temp = n;

while(temp>0)

{

rem = temp%10;

rev = (rev\*10)+rem;

temp = temp/10;

}

if(rev==n)

return 0;

else

return 1;

}

3.Write a C++ Program to find Cube of a Number using function.

Code-

**#include <iostream>**

**using** **namespace** std;

**int** cubeNum(**int**);

**int** main(){

**int** n, cube;

// Asking for input

cout << "Enter a number: ";

cin >> n;

// Calling out function

cube = cubeNum(n);

// Displaying output

cout << "Cube of " << n << " is: " << cube << endl;

**return** 0;

}

**int** cubeNum(**int** x){

**return** (x \* x \* x);

}

4. Write a C++ Program to Swap two numbers and characters using call by value.

Code-

#include<iostream>

using namespace std;

void swap(int,int);

void swapc(char,char);

int main()

{

char x,y;

int a,b;

cout<<"\nEnter 1st character :: ";

cin>>x;

cout<<"\nEnter 2nd character :: ";

cin>>y;

cout<<"\nEnter 1st integer :: ";

cin>>a;

cout<<"\nEnter 2nd integer :: ";

cin>>b;

cout<<"\nBefore Swapping, Value of Characters :: \n\tx = "<<x<<"\ty = "<<y<<"\n";

cout<<"\nBefore Swapping, Value of Integers :: \n\tA = "<<a<<"\tB = "<<b<<"\n";

swap(a,b);

swapc(x,y);

cout<<"\nOutside Function After Swapping, Value of Characters :: \n\tx = "<<x<<"\ty = "<<y<<"\n";

cout<<"\nOutside Function After Swapping, Value of Integers :: \n\tA = "<<a<<"\tB = "<<b<<"\n";

}

void swap(int a,int b)

{

int c;

c=a;

a=b;

b=c;

cout<<"\nInside Function After Swapping, Value of Integers :: \n\tA = "<<a<<"\tB = "<<b<<"\n";

}

void swapc(char x,char y)

{

char z;

z=x;

x=y;

y=z;

cout<<"\nInside Function After Swapping, Value of Characters :: \n\tx = "<<x<<"\ty = "<<y<<"\n";

}