Wednesday, 19 January 2022

Q1 – Declare a class employee.

Solution - ../Programs/00\_Class.cpp

*// Declare a class for Employees*

#include <iostream>

using namespace std ;

class *Employee* {

    int id ;

    char name[30] ;

    float ctc ;

} ;

int main () {

*Employee* roy, agr ;

    return 0;

}

Thursday, 20 January 2022

Q1 Program to find the largest among three integer.

Solution - ../Program/04\_Practice.cpp

#include <iostream>

using namespace std;

int max(int a, int b, int c)

{

    if (a > b && a > c)

    {

        return a;

    }

    else if (b > a && b > c)

    {

        return b;

    }

    else

    {

        return c;

    }

}

int main()

{

    int a, b, c;

    cout << "Enter a, b, c " << endl;

    cin >> a >> b >> c;

    cout << "Largest number is " << max(a, b, c) << endl;

    return 0;

}

Q2 Write a Program to find the sum of natural numbers.

Solution – ../Program/04\_Practice.cpp

#include <iostream>

using namespace std;

int sumN(int n)

{

    if (n == 0)

    {

        return 0;

    }

    else

    {

        return n + sumN(n - 1);

    }

}

int main()

{

    int n;

    cout << "Enter no. of terms " << endl;

    cin >> n;

    cout << "Sum of " << n << " natural number is " << sumN(n) << endl;

    return 0;

}

Q3 WAP to check if a number is prime or not.

Solution – ../Program/04\_Practice.cpp

#include <iostream>

using namespace std;

int isPrime(int a)

{

    if (a == 1)

    {

        return 0;

    }

    for (int i = 2; i <= a / 2; i++)

    {

        if (a % i == 0)

        {

            return 0;

        }

    }

    return 1;

}

int main()

{

    int n;

    cout << "Enter Number " << endl;

    cin >> n;

    cout << "Isprime : " << isPrime(n) << endl;

    return 0;

}

Q4 WAP to display the Fibonacci series.

Solution – ../Program/04\_Practice.cpp

#include <iostream>

using namespace std;

int Fibonnaciterm(int n)

{

    if (n == 1)

    {

        return 0;

    }

    else if (n == 2)

    {

        return 1;

    }

    else

    {

        return Fibonnaciterm(n - 1) + Fibonnaciterm(n - 2);

    }

}

int main()

{

    int n;

    cout << "Enter n " << endl;

    cin >> n;

    for (int i = 1; i <= n; i++)

    {

        cout << Fibonnaciterm(i) << " "  ;

    }

    return 0;

}

Q5 – What is a class?

Answer – A class is a collection of different data and functions.

Q7 – What is an object?

Answer - An object is an instance of class referring to a real entity.

Friday, 21 January 2022

Q1 – Class to represent the details of a n students.

Solution –

#include <iostream>

using namespace std;

class student

{

private:

int roll;

float percentage;

public:

void input();

void output();

// char grade(float a);

char grade();

};

void student ::input()

{

cout << "Enter roll " << endl;

cin >> roll;

cout << "Enter Percentage " << endl;

cin >> percentage;

}

char student ::grade()

{

if (percentage >= 90)

{

return 'A';

}

else if (percentage >= 80)

{

return 'B';

}

else if (percentage >= 70)

{

return 'C';

}

else

{

return 'D';

}

}

void student ::output()

{

cout << "Roll - " << roll << " Percentage - " << percentage << endl;

cout << "Grade - " << this->grade() << endl;

}

int main()

{

int n;

cout << "Enter the no. of students " << endl;

cin >> n;

student S[n];

for (int i = 0; i < n; i++)

{

cout << "Student " << i + 1 << " : " << endl;

S[i].input();

}

for (int i = 0; i < n; i++)

{

cout << "Student " << i + 1 << " : " << endl;

S[i].output();

}

return 0;

}

Q2 – Class to represent the details of 2 students

Solution –

#include <iostream>

using namespace std;

class student

{

private:

int roll;

float percentage;

public:

void input();

void output();

char grade();

} harsh, utkarsh;

void student ::input()

{

cout << "Enter roll " << endl;

cin >> roll;

cout << "Enter Percentage " << endl;

cin >> percentage;

}

char student ::grade()

{

if (percentage >= 90)

{

return 'A';

}

else if (percentage >= 80)

{

return 'B';

}

else if (percentage >= 70)

{

return 'C';

}

else

{

return 'D';

}

}

void student ::output()

{

cout << "Roll - " << roll << " Percentage - " << percentage << endl;

cout << "Grade - " << this->grade() << endl;

}

int main()

{

harsh.input();

utkarsh.input();

harsh.output();

utkarsh.output();

return 0;

}

Q3 – WAP to represent the details of a student.

Solution –

#include <iostream>

using namespace std;

class student

{

private:

int roll;

float percentage;

public:

void input();

void output();

// char grade(float a);

char grade() ;

};

void student ::input()

{

cout << "Enter roll " << endl;

cin >> roll;

cout << "Enter Percentage " << endl;

cin >> percentage;

}

char student ::grade()

{

if (percentage >= 90)

{

return 'A';

}

else if (percentage >= 80)

{

return 'B';

}

else if (percentage >= 70)

{

return 'C';

}

else

{

return 'D';

}

}

void student ::output()

{

cout << "Roll - " << roll << " Percentage - " << percentage << endl;

cout << "Grade - " << this->grade() << endl;

}

int main()

{

student harsh;

harsh.input();

harsh.output();

return 0;

}

Q4 – What are access specifiers in C++?

Answer –

Access Specifiers define how the members of a class can be accessed.

There are three access specifiers in cpp –

1. Public : Members are accessible from outside the class.
2. Private : Members cannot be accessed or viewed from outside the class scope.
3. Protected : Members cannot be accessed or viewed from outside the class however

Can be accessed from inherited class.

Example –

class MyClass {  
  **public:**    // Public access specifier  
    int x;   // Public attribute  
  **private:**   // Private access specifier  
    int y;   // Private attribute  
};  
  
int main() {  
  MyClass myObj;  
  myObj.x = 25;  // Allowed (public)  
  myObj.y = 50;  // Not allowed (private)  
  return 0;  
}

Q5 – What is an object?

Answer –

An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. Defining Class and Declaring Objects. A class is defined in C++ using keyword class followed by the name of class

Here,

Class student {

int data ;

}s1 ;

Student is the class , and s1 is the object.

Q6 – How to define a member function?

Answer –

There are two ways to define a member :

1. Inside the class declaration : We can define a function inside a class along with its

declaration. We can define a function inside a class if the function definition is

less than 4 lines. However it is not a standard practice to define a function inside a class

definition.

Example –

Class student {

int roll ;

void input () {

cout << “Enter Roll” << endl;

cin >> roll;

}

}

Here, the function input is defined within the class declaration, It’s wont give an

Error since there are only two line of code in input function.

1. Outside the class definition : We can Define a function outside the class definition using the

Scope resolution operator. In this method we declare the function with appropriate arguments and define it outside the class block.

Syntax : <return type> <class name > :: <function name> () {

// code goes here

}

This is a standard way of defining member function.

Thursday, 27 January 2022

Q1 What is structure ?

Answer – Structure is a user defined data type used to collect different types of data member. In c++, We have member function for structure too.

The size of a structure is equal to the sum of the size of its data members, however an empty structure takes up one byte of memory. Structures in c++ has two access specifiers, Public and Private. By default, the access specifiers of structure is set to public.

Q2 – What is union ?

Answer – Unions are user defined data type which uses a common memory location for its data members. The size of an union data type is equal to the size of largest data member.

Unions are used for efficient management of space. We can initialise a data member during declaration, but at max only one initialisation is allowed in union.

Q3- What is enum ?

Answer – An enumeration is a user defined data type consisting of set of named constant called enumerators.

Q4- Difference between a class and a structure.

Answer –

|  |  |
| --- | --- |
| Class   * A class in c++ can be defined as a collection of related variables and function encapsulated in a single structure. * Keyword : class * Three access specifiers : public, private and protected. * Default access specifiers : Private. * Has features like data abstraction, inheritance etc. * Used for large amount of data. | Structure   * A structure can be referred to as a user defined data type possessing its own operations. * Keyword : struct * Two access specifiers : public and private. * Default access specifiers : Public * Grouping of data * Used for smaller amount of data. |

Q5 – Difference between structure and union.

Answer –

|  |  |
| --- | --- |
| Structure   * A structure can be referred to as a user defined data type possessing its own operations. * Keyword : struct * Two access specifiers : public and private. * Default access specifiers : Public * Grouping of data * Used for smaller amount of data. | Union   * A union is a user defined data type, with its data member sharing a common memory location. * Keyword : union * No access specifiers. * No default access specifiers. * No member function. * Max one initialisation allowed. * Used for better memory management |

Q6 – WAP to display to display the student info -

#include <iostream>

using namespace std;

struct student

{

private:

int roll;

char name[20];

public:

void setdata();

void display();

};

void student ::setdata()

{

cout << "Enter name : ";

cin >> name;

cout << "Enter roll : ";

cin >> roll;

}

void student ::display()

{

cout << "Name : " << name << " Roll : " << roll << endl;

}

union stud

{

int registration;

int roll;

};

enum day

{

mon,

tue = 8,

wed,

thur = 5,

fri,

sat,

sun

};

int main()

{

student S1;

S1.setdata();

S1.display();

stud S2;

cout << "Enter registration no. :: ";

cin >> S2.registration;

cout << "Resgistration number is " << S2.registration << endl;

cout << "Enter Roll : ";

cin >> S2.roll;

cout << "Roll is : " << S2.roll << endl;

day x, y, z;

x = mon;

y = fri;

z = wed;

cout << x << endl;

cout << x + 3 << endl;

cout << y << endl;

cout << z << endl;

return 0;

}