## **CHAUDHARY HAMDAN**

**1905387**

**OOP LAB-2**

**Date : 07-08-2020**

1. WAP to display the message "hello" followed by your name on screen.

#include <iostream>

using namespace std;

int main()

{

cout<<"Hello KIITian";

return 0;

}

OUTPUT:



1. Create a class which stores name, roll number and total marks for a student. Input the data for a student and display it.

#include <iostream>

using namespace std;

class student

{

string name;

int roll;

int marks;

public:

void getdata()

{

cout<<"Enter Name: ";

cin>>name;

cout<<"Enter Roll Number: ";

cin>>roll;

cout<<"Enter Marks: ";

cin>>marks;

}

void display()

{

cout<<"========================="<<endl;

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

cout<<"Roll Number: "<<roll<<endl;

cout<<"Marks: "<<marks<<endl;

}

};

int main()

{

student ob;

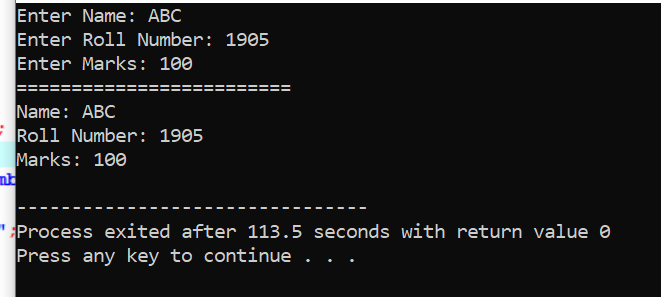
ob.getdata();

ob.display();

return 0;

}

OUTPUT:



1. Modify the program ii) to store marks in 5 subjects. Calculate the total marks and percentage of a student and display it.

#include <iostream>

using namespace std;

class student

{

string name;

int roll;

int marks[5];

int tmarks=0;

double percentage;

public:

void getdata()

{

cout<<"Enter Name: ";

cin>>name;

cout<<"Enter Roll Number: ";

cin>>roll;

cout<<"Enter Marks of five subjects: ";

for(int i=0; i<5; i++)

{

cout<<"Enter Marks of Subject "<<i+1<<" : ";

cin>>marks[i];

tmarks = (tmarks + marks[i]);

}

}

void display()

{

cout<<"========================="<<endl;

cout<<"Student's Name: "<<name<<endl;

cout<<"Roll Number: "<<roll<<endl;

for(int i=0; i<5; i++)

{

cout<<"Marks of Subject "<<i<<" : ";

cout<<marks[i]<<endl;

}

cout<<"Total Marks: "<<(tmarks)<<endl;

cout<<"Percentage : "<<(tmarks/5)<<"%"<<endl;

}

};

int main()

{

student ob;

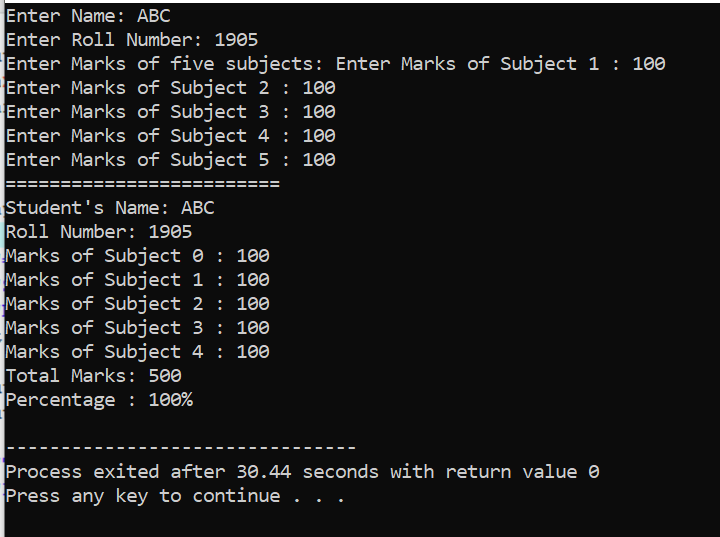
ob.getdata();

ob.display();

return 0;

}

OUTPUT:



1. Create a class complex which stores real and imaginary part of a complex number. Input 10 complex numbers and display them.

#include <iostream>

using namespace std;

class complex

{

int real;

int img;

public:

void getdata()

{

cout<<"Enter Real Part: ";

cin>>real;

cout<<"Enter Imaginary Part: ";

cin>>img;

}

void display()

{

static int n=1;

cout<<"Number "<<n++<<" : "<<real<<"+"<<img<<"i"<<endl;

}

};

int main()

{

complex ob[10];

for(int i=0; i<10; i++)

{

ob[i].getdata();

}

cout<<"========================="<<endl;

for(int i=0; i<10; i++)

{

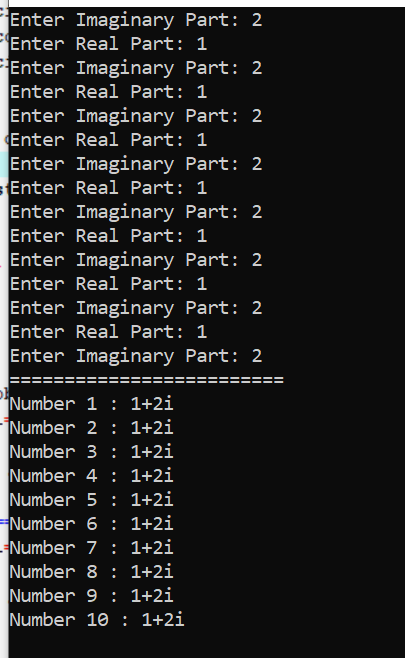
ob[i].display();

}

return 0;

}

OUTPUT:



1. Create a class distance which stores a distance in feet and inches. Input 2 distance values

in objects, add them, store the resultant distance in and object and display it.

[Write the above program in two ways.

a) store the resultant distance in the calling object:C3.add(C1,C2)

b) return the resultant object C3=C1.add(C2)

#include <iostream>

using namespace std;

class dist

{

double feet;

double inches;

public:

void getdata()

{

cout<<"Enter Feet and Inches: ";

cin>>feet>>inches;

}

void display()

{

cout<<feet<<"'"<<inches<<"''"<<endl;

}

void add(dist a, dist b)

{

inches=a.inches+b.inches;

feet=a.feet+b.feet;

}

dist add(dist b)

{

dist temp;

temp.inches=inches+b.inches;

temp.feet=feet+b.feet;

return temp;

}

};

int main()

{

dist c1,c2,c3,c4;

c1.getdata();

c2.getdata();

cout<<"Distance 1: ";

c1.display();

cout<<"Distance 2: ";

c2.display();

c3.add(c1,c2);

cout<<"Output for c3.add(c1,c2): "<<endl;

c3.display();

c4=c1.add(c2);

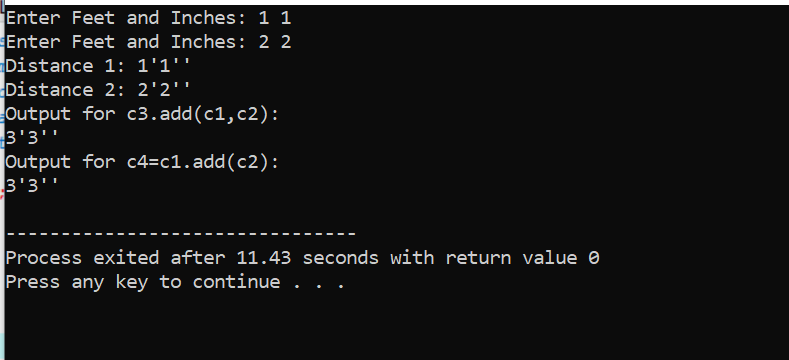
cout<<"Output for c4=c1.add(c2): "<<endl;

c4.display();

return 0;

}

OUTPUT:



1. Create a class which stores id, name, age and basic salary of an employee. Input data for

n number of employees. Calculate the gross salary of all the employees and display it

along with all other details in a tabular form.

[Gross salary= Basic salary + DA + HRA,

DA = 80% of Basic salary HRA=10%

of Basic salary ]

#include <iostream>

using namespace std;

class employee

{

string name;

int age;

int basic,DA,HRA;

double gross;

public:

void getdata()

{

cout<<"Enter Name: ";

cin>>name;

cout<<"Enter Age: ";

cin>>age;

cout<<"Enter Basic salary: ";

cin>>basic;

DA=0.8\*basic;

HRA=0.1\*basic;

gross=basic+DA+HRA;

}

void display()

{

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

cout<<"Age: "<<age<<endl;

cout<<"Basic salary = "<<basic<<endl;

cout<<"Gross salary = "<<gross<<endl;

}

};

int main()

{

int n=0;

cout<<"Enter number of employee: ";

cin>>n;

employee ob[n];

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

{

cout<<"Enter Employee "<<i+1<<" details"<<endl;

ob[i].getdata();

}

cout<<"========================="<<endl;

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

{

cout<<"Employee "<<i+1<<endl;

ob[i].display();

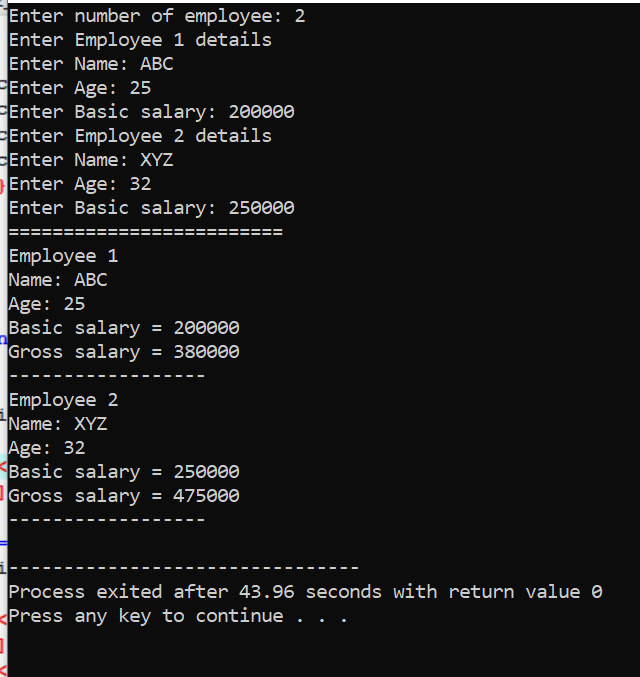
cout<<"------------------"<<endl;

}

return 0;

}

OUTPUT:



1. Create a class which stores x and y coordinates of a point. Calculate distance between two given points and display it.

#include <iostream>

#include<math.h>

using namespace std;

class dist

{

int x,y;

public:

void getdata()

{

cout<<"Enter x and y coordinates : ";

cin>>x>>y;

}

void display()

{

cout<<"("<<x<<","<<y<<")"<<endl;

}

double add(dist a, dist b)

{

return sqrt(pow(b.y - a.y, 2) + pow(b.x - a.x, 2));

}

};

int main()

{

dist c1,c2;

c1.getdata();

c2.getdata();

cout<<"Coordinate 1: ";

c1.display();

cout<<"Coordinate 2: ";

c2.display();

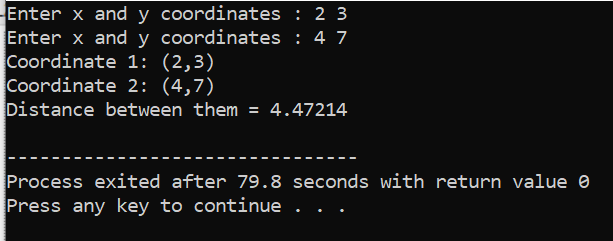
c1.add(c1,c2);

cout<<"Distance between them = "<<c1.add(c1,c2)<<endl;

return 0;

}

OUTPUT:



1. nth fibbonacci

#include <iostream>

using namespace std;

int fib(int n)

{

if(n == 1)

return 1;

else if(n==2)

return 1;

else

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

}

int main()

{

int n;

cin>>n;

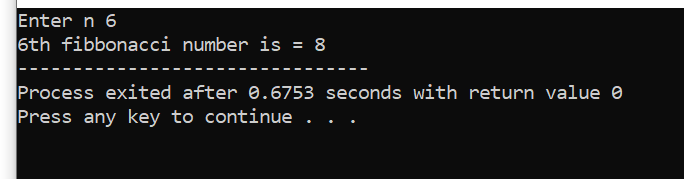
int ans=fib(n);

cout<<ans;

return 0;

}

OUTPUT :



1. Strong Number

#include<iostream>

using namespace std;

int fact(int n)

{

int x = 1;

for(int i=1;i<=n;i++)

x = x\*i;

return x;

}

int strong(int n)

{

int t,r,ans=0;

t = n;

while(t)

{

r = t%10;

ans = ans + fact(r);

t = t/10;

}

if(ans == n)

return 1;

else

return 0;

}

int main()

{

cout << "Strong numbers between 1 and 1000 are "<<endl;

for(int i=1;i<1001;i++)

{

if(strong(i))

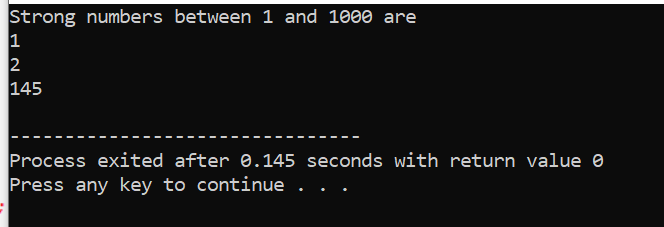
cout <<i<<endl;

}

return 0;

}

OUTPUT :



1. Perfect Number

#include<iostream>

using namespace std;

int main()

{

int sum;

cot<<"Perfect numbers between 1 to 1000 are : ";

for(int i=1;i<=1000;i++)

{

sum=0;

for(int j=1;j<=(i/2);j++)

{

if(i%j == 0)

{

sum+=j;

}

}

if(sum==i) {

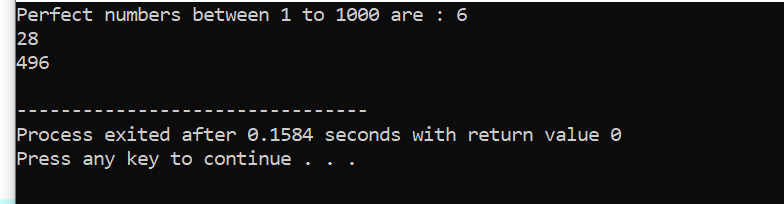
cout << i << endl;

}

}

}

OUTPUT :



1. Prime Factor

#include<iostream>

using namespace std;

int main()

{

int n,prime;

cout<<"Enter n :";

cin >> n;

cout<<"Prime Factors of "<<n<<" are: ";

for(int i=2;i<n+1;i++)

{

if(n%i==0)

{

prime = 1;

for(int j=2; j<=i/2; j++)

{

if(i%j==0)

{

prime = 0;

break;

}

}

if(prime==1)

{

cout << i << " ";

}

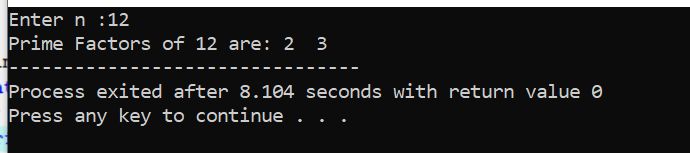
}

}

return 0;

}

OUTPUT :



1. Print Bigger Number

#include<iostream>

using namespace std;

int main()

{

int a,b;

cout<<"Enter two numbers : ";

cin >> a >> b;

int n = 0;

while(a>0 || b>0)

{

--a;

--b;

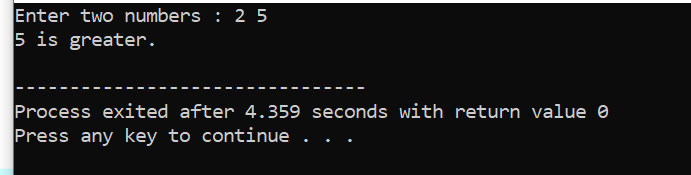
n += 1;

}

cout << n << " is greater. " << endl;

}

OUTPUT :



1. Check if divisible by 5 or not without using % operator

#include<iostream>

using namespace std;

int main()

{

int n;

cout <<"Enter n : ";

cin >> n;

int a=n;

while(a>0)

{

a=a-5;

}

if(a==0)

cout << n << " is divisible by 5"<< endl;

else

cout << n << " is NOT divisible by 5"<< endl;

}

OUTPUT :

