DSA0136OBJECT ORIENTED PROGRAMMING WITH C++

Name:ch.vamsi krishna

Reg:192110482

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

WAP in C++ to print the name of student by creating class- student. If number and name is passed while creating an object of student class then the name should be “unknown” while creating object to the class.

PROGRAM-

#include <iostream>

using namespace std;

class student

{

int number;

char \*name;

public:

student();

student(int,char\*);

void display();

};

student::student()

{

number=0;

char \*name="Unknown";

}

student::student(int x,char \*y)

{

number=x;

name=y;

}

void student::display()

{

cout<<"\nenter the number:"<<number;

cout<<"\nenter the name:"<<name;

}

int main()

{

student s(10,"Sachin");

student s1(7,"Mahi");

student s2(18,"Virat");

s.display();

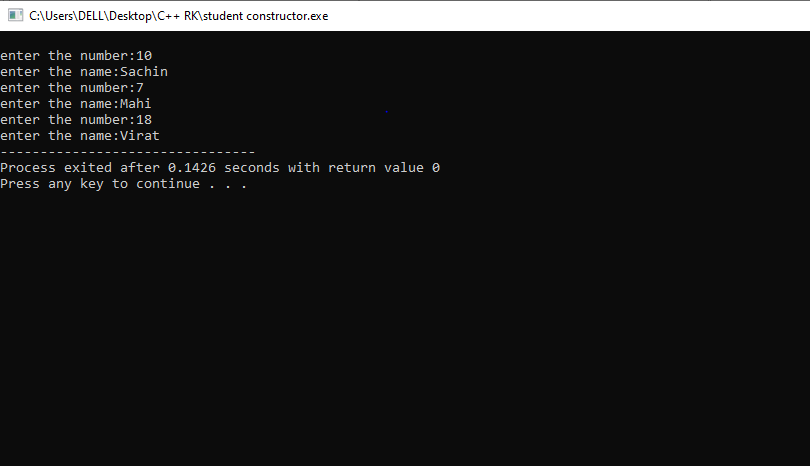
s1.display();

s2.display();

return 0;

}

OUTPUT-



2. WAP in C++ to create a class rectangle with parameters length and breadth and function to calculate the area of rectangle.

This class contains 3 constructors-(i)->without parameters,(ii)->with 2 parrameters,

(iii)->with 1 parameter

PROGRAM-

#include <iostream>

using namespace std;

class rectangle

{

int l,b,a;

public:

rectangle();

rectangle(int,int);

rectangle(int);

void calculate();

};

rectangle::rectangle()

{

l=0;

b=0;

};

rectangle::rectangle(int x,int y)

{

l=x;

b=y;

};

rectangle::rectangle(int x)

{

l=x;

b=4;

};

void rectangle::calculate()

{

a=l\*b;

cout<<"\nthe area of rectangle is:"<<a;

};

int main()

{

rectangle r(0,0);

rectangle r1(7,5);

rectangle r2(5);

r.calculate();

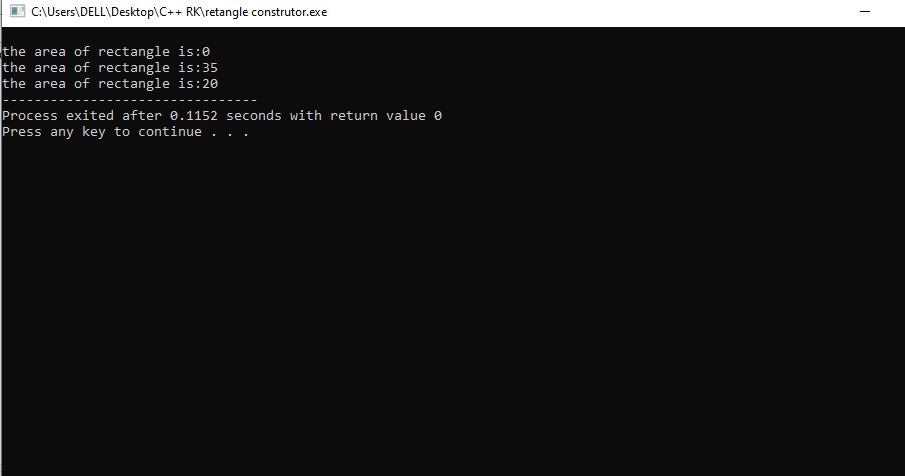
r1.calculate();

r2.calculate();

return 0;

}

OUTPUT-



3. WAP in C++ to create class Addamount and with 2 constructors one with only amount and one with incrementing the amount.

Then finally display the final amount with addition in the second constructor.

PROGRAM-

#include<iostream>

using namespace std;

class AddAmount

{

int amount;

public:

AddAmonut()

{

amount = 50;

}

AddAmount(int x)

{

amount = x+50;

}

void display()

{

cout<<"amount is:"<<amount;

}

};

int main()

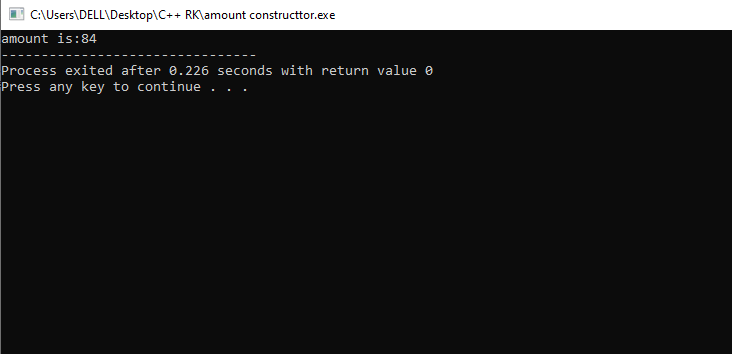
{

AddAmount a(34);

a.display();

}

OUTPUT-



WAP in C++ to print different types of data with same function name. class name-printNumber

PROGRAM-

#include<iostream>

using namespace std;

class printnumber

{

int a;

char b;

float c;

double d;

bool e;

public:

int printn(int x)

{

a = x;

return x;

}

char printn(char y)

{

b = y ;

return y;

}

float printn(float z)

{

c = z;

return z;

}

double printn(double s)

{

d = s;

return s;

}

bool printn(bool f)

{

e = f;

return f;

}

};

int main()

{

printnumber p;

cout<<"\n integer is:"<<p.printn(3);

cout<<"\n charactor is:"<<p.printn('k');

cout<<"\n float is:"<<p.printn(3.5);

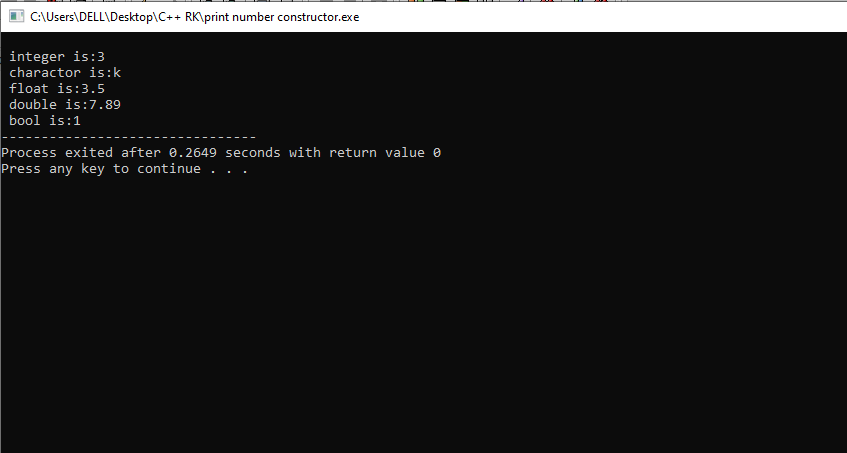
cout<<"\n double is:"<<p.printn(7.89);

cout<<" \n bool is:"<<p.printn(true);

return 0;

}

OUTPUT-



WAP in C++

PROGRAM-

#include<iostream>

using namespace std;

class print{

public:

void output(int a, char b[20])

{

cout<<"Integer is "<<a<<endl;

cout<<"Character is "<<b<<endl;

}

void output(char b[20], int a)

{

cout<<"Character is "<<b<<endl;

cout<<"Integer is "<<a<<endl;

}

};

int main()

{

print obj;

obj.output("kisore", 10);

obj.output(10, "kishore");

}

OUTPUT-

