Function Overloading

#include <iostream>

using namespace std;

class functionoverloading

{

public:

void print(int i)

{

cout << "Printing int: " << i << endl;

}

void tisha(double f)

{

cout << "Printing float: " << f << endl;

}

double tisha(double a, double b)

{

cout << "Printing value of a: " << a << endl;

cout << "Printing value of b: " << b << endl;

}

void tisha(char\* c)

{

cout << "Printing character: " << c << endl;

}

};

int main(void)

{

functionoverloading pd;

pd.tisha(5);

pd.tisha(500.263);

pd.tisha(70.3, 89.6);

pd.tisha("Tisha");

return 0;

}

//function overloading same name parameter(int, double etc) or return type(void, double etc) change

OUTPUT:



Operator Overloading

#include<iostream>

using namespace std;

class complex

{

public:

complex operator++()

{

cout<<"Operator overloading ++"<<endl;

}

complex operator--()

{

cout<<"Operator overloading --"<<endl;

}

};

int main()

{

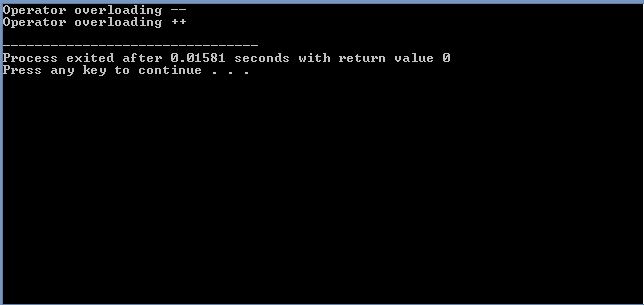
complex obj1;

--obj1;

++obj1;

}

OUTPUT:



Pure Virtual Function

#include<iostream>

using namespace std;

class abc

{

public:

virtual void func()=0; //pure virtual function

};

class Tisha:public abc

{

public:

void func()

{

cout<<"This is function of Tisha"<<endl;

}

};

class Tisah1:public abc

{

public:

void func()

{

cout<<"This is function of Tisha1"<<endl;

}

};

int main()

{

//abc ob1; aita lekha jabe na. pure virtual function a aita error

Tisha ob2;

Tisah1 ob3;

ob2.func();

ob3.func();

}

//pure virtual function je class a thake oi class ke abstract class bole

OUTPUT:



Multi-level inheritance

#include<iostream>

using namespace std;

class tisha

{

public:

int tisha1=100;

void show()

{

cout<<"This function is from tisha"<<endl;

}

};

class tisharjamai:public tisha

{

public:

int tisha2=200;

void show()

{

cout<<"This function is from tisha-r-jamai"<<endl;

}

};

class priyam:public tisharjamai

{

public:

void showvalue()

{

cout<<"This value is from tisha class: " << tisha1 << endl;

cout<<"This value is from tisha-r-jamiai class: " << tisha2 << endl;

}

};

int main()

{

tisha ob1;

tisharjamai ob2;

priyam ob3;

ob1.show();

ob2.show();

ob3.showvalue();

}

OUTPUT:

