**LAB # 8**

**OBJECT:**

To understand member function overriding.

**SOURCE CODE:**

#include <iostream>

#include <process.h>

using namespace std;

class Stack{

protected:

enum { MAX = 3 };

int st[MAX];

int top;

public:

Stack()

{ top = -1; }

void push(int var)

{ st[++top] = var; }

int pop()

{ return st[top--]; }

class Stack2 :: public Stack{

public:

void push(int var) //put number on stack

{

if(top >= MAX-1)

{ cout << “\nError: stack is full”;

exit(1); }

Stack::push(var);

}

int pop()

{

if(top < 0)

{ cout << “\nError: stack is empty\n”;

exit(1); }

return Stack::pop();

}

};

int main()

{

Stack2 s1;

s1.push(11);

s1.push(22);

s1.push(33);

cout << endl << s1.pop();

cout << endl << s1.pop();

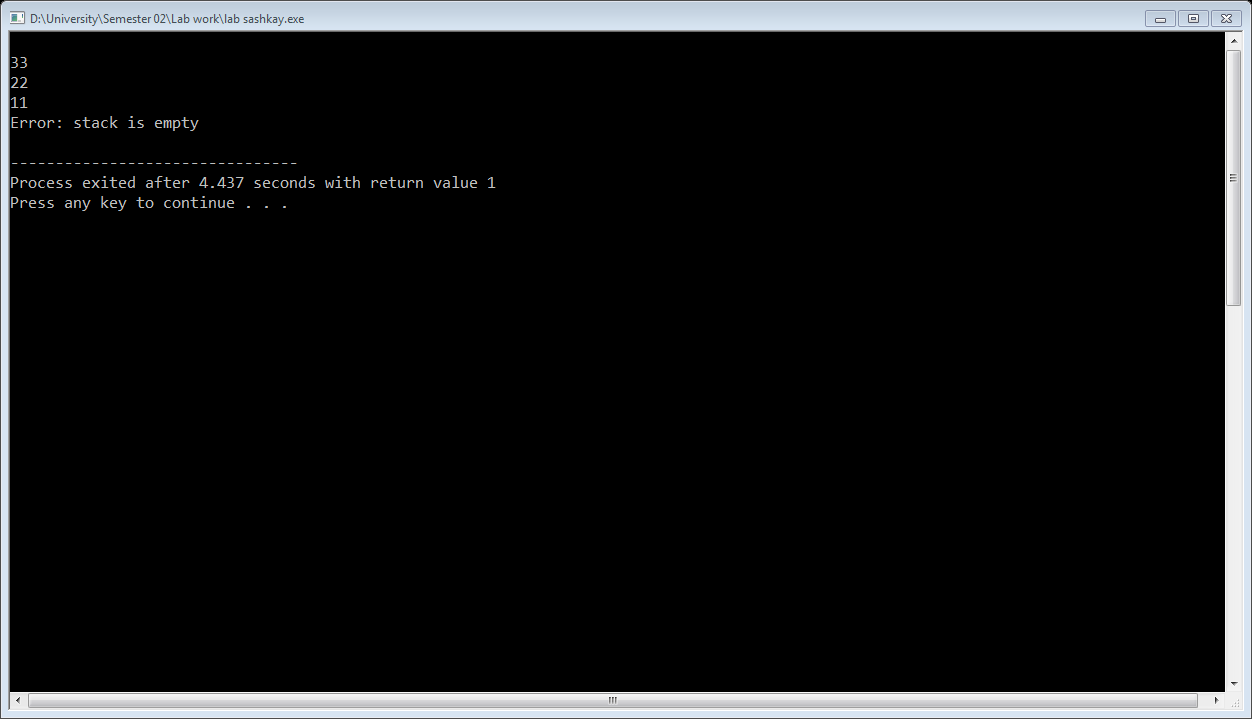
cout << endl << s1.pop();

cout << endl << s1.pop();

cout << endl;

return 0;

}



**EXERCISE:**

Imagine a publishing company that markets both book and audiocassette versions

of its works. Create a class publication that stores the title (a string) and price

(type float) of a publication. From this class derive two classes: book, which adds

a page count (type int); and tape, which adds a playing time in minutes (type

float). Each of these three classes should have a getdata() function to get its data

from the user at the keyboard, and a putdata() function to display its data. Write a

main() program to test the book and tape classes by creating instances of them,

asking the user to fill in data with getdata(), and then displaying the data with

putdata().

**SOURCE CODE:**

#include<iostream>

#include<string>

using namespace std;

class Publication{

private:

string title;

float price;

public:

void getData()

{

cout<<"\nEnter Title: "; cin>>title;

cout<<"Enter Price: $"; cin>>price;

}

void putData()

{ cout<<"\nTitle: "<<title<<"\nPrice: "<<price; }

};

class Book : public Publication{

private:

int pages;

public:

void getData()

{ Publication::getData(); cout<<"Enter Pages: "; cin>>pages; }

void putData()

{ Publication::putData(); cout<<"\nPages: "<<pages; }

};

class Tape : public Publication{

private:

float minutes;

public:

void getData()

{ Publication::getData(); cout<<"Enter Minutes: "; cin>>minutes; }

void putData()

{ Publication::putData(); cout<<"\nMinutes: "<<minutes; }

};

int main()

{

Book b;

Tape t;

b.getData();

t.getData();

b.putData();

cout<<endl;

t.putData();

cout<<endl;

return 0;

}