Question 2:

#include<iostream>

#include<string>

using namespace std;

class Book

{

private:

int books;

public:

Book(){} //default constructor

//Accessors

void set\_book(int books)

{

this->books = books;

}

//Mutators

int get\_book()

{

return books;

}

};

class Student

{

private:

int hours, fine, no\_of\_book; //private member variables

public:

Student(){} //default constructor

//Accessors declarations

void set\_hours(int hours);

void set\_fine(int);

void set\_no\_of\_book(int no\_of\_book);

//mutators declarations

int get\_hours();

int get\_fine();

int get\_no\_of\_book();

void max\_book();

void display();

};

int main()

{

int n,n1;

Book obj; //book object

Student obj2; //student object

obj.set\_book(15);

cout << "welcome to Fast uni Library" << endl;

cout << "how many books you want" << endl;

cin >> n;

obj2.set\_no\_of\_book(n);

obj2.max\_book();

cout << "late hours of submission the book" << endl;

cin >> n1;

obj2.set\_hours(n1);

obj2.get\_fine();

obj2.display();

}

//Accessors defintions

void Student::set\_fine(int fine = 5)

{

this->fine = fine;

}

void Student::set\_hours(int hours)

{

this->hours = hours;

}

void Student::set\_no\_of\_book(int no\_of\_book)

{

this->no\_of\_book = no\_of\_book;

}

//Muatators definitions

int Student ::get\_hours()

{

return hours;

}

int Student::get\_fine()

{

set\_fine(5);

fine = fine \* hours;

return fine;

}

int Student::get\_no\_of\_book()

{

return no\_of\_book;

}

void Student::display()

{

cout << "your total fine is : " << get\_fine() << endl;

}

void Student::max\_book()

{

if (get\_no\_of\_book() > 10)

{

cout << "Permission Denied. number of books more than 10 is not allowed" << endl;

cout << "try again" << endl;

}

while (get\_no\_of\_book() > 10)

{

int n;

cin >> n;

set\_no\_of\_book(n);

get\_no\_of\_book();

if (get\_no\_of\_book() > 10)

{

cout << "Permission Denied. number of books more than 10 is not allowed" << endl;

cout << "try again" << endl;

}

}

}

Question 3:

#include <iostream>

using namespace std;

class shop {

private:

int id, salary;

string name, status;

int post;

public:

void set() {

cout << "Enter your name: ";

cin >> name;

cout << "Enter your status(working/not working): ";

cin >> status;

cout << "Enter 1 for Owner" << endl;

cout << "Enter 2 for Salesman" << endl;

cout << "Enter 3 for Manager" << endl;

cout << "Enter your post: ";

cin >> post;

cout << "Enter your ID: ";

cin >> id;

cout << "Enter your salary: ";

cin >> salary;

}

void get() {

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

cout << "Status: " << status << endl;

cout << "ID: " << id << endl;

cout << "Salary: " << salary << endl;

if (post == 1) {

cout << "Post: " << "Owner" << endl;

}

else if (post == 2) {

cout << "Post: " << "Salesman" << endl;

}

else {

cout << "Post: " << "Manager" << endl;

}

}

void operator -() {

cout << "Enter your updated status(working/not): ";

cin >> status;

}

void operator \*() {

cout << "Enter the ID of the person to be increamented: ";

int temp;

cin >> temp;

if (temp == id) {

int result = salary + (salary \* 10 / 100);

salary = result;

}

}

friend void operator >>(istream &in, shop &c) {

cout << "Enter your name: ";

in >> c.name;

cout << "Enter your status(working/not working): ";

in >> c.status;

cout << "Enter 1 for Owner" << endl;

cout << "Enter 2 for Salesman" << endl;

cout << "Enter 3 for Manager" << endl;

cout << "Enter your post: ";

in >> c.post;

cout << "Enter your ID: ";

in >> c.id;

cout << "Enter your salary: ";

in >> c.salary;

}

friend void operator <<(ostream &out, shop &b) {

cout << endl;

out << "Name: " << b.name << endl;

out << "Status: " << b.status << endl;

out << "ID: " << b.id << endl;

out << "Salary: " << b.salary << endl;

if (b.post == 1) {

cout << "Post: " << "Owner" << endl;

}

else if (b.post == 2) {

cout << "Post: " << "Salesman" << endl;

}

else {

cout << "Post: " << "Manager" << endl;

}

}

void operator ++() {

salary = salary + 5000;

cout <<"new salary is : " << salary;

}

void operator --() {

salary = salary - 5000;

}

void operator ==(shop a) {

}

void operator !=(shop a) {

}

};

int main() {

shop a[5];

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

a[i].set();

a[i].get();

++a[i];

--a[i];

cin >> a[i];

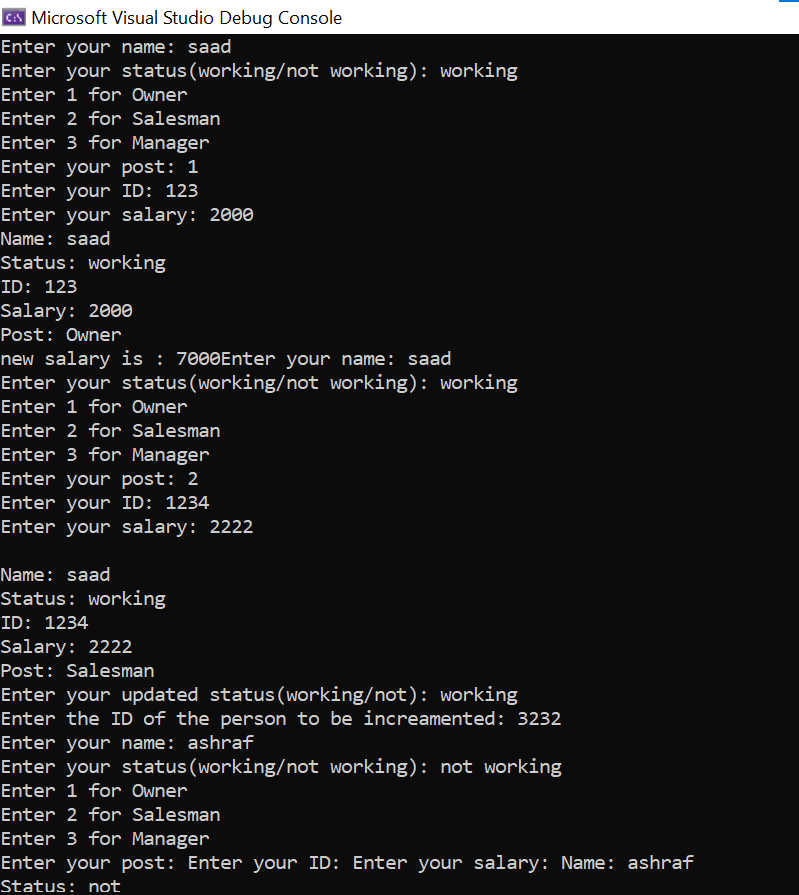
cout << a[i];

-a[i];

\*a[i];

}

}



Question 4:

#include<iostream>

using namespace std;

class Base

{

private:

int Base3;

friend class Friend\_Class;

public:

int Base1, Base2;

Base(){

Base1 = 0;

Base2 = 0;

Base3 = 0;

}

Base(int Base1, int Base2, int Base3){

this->Base1 = Base1;

this->Base2 = Base2;

this->Base3 = Base3;

}

void set(int Base1, int Base2, int Base3){

this->Base1

= Base1;

this->Base2 = Base2;

this->Base3 = Base3;

}

void print(){

cout << " " << "Base 1 is : " << Base1 << endl;

cout << " " << "Base 2 is : " << Base2 << endl;

cout << " " << "Base 3 is : " << Base3 << endl;

}

};

class Derived :public Base

{

private:

int Derived1;

int Derived2;

public:

Derived() :Base(1, 1, 1){

Derived1 = 1, Derived2 = 2;

}

Derived(int Derived1, int Derived2){

this->Derived1 = Derived1;

this->Derived2 = Derived2;

}

void set(int Derived1, int Derived2){

this->Derived1 = Derived1;

this->Derived2 = Derived2;

}

void print(){

cout << " " << "Derived1 : " << Derived1 << endl;

cout << " " << "Derived2 : " << Derived2 << endl;

}

friend class Friend1\_Class;

};

class Friend\_Class

{

Base b1;

public:

Friend\_Class() {}

void accessMethod(){

cout << "......Access from Base Class by making friend class......\n";

b1.set(12, 31, 45);

cout << " " << "Base1 : " << b1.Base1 << endl;

cout << " " << "Base2 : " << b1.Base2 << endl;

cout << " " << "Base3 : " << b1.Base3 << endl;

}

friend class Friend2\_Class;

};

class Friend1\_Class

{

Derived d;

public:

void accessMethod(){

cout << "......Access from Derived Class by making friend class......\n";

d.set(34, 65);

cout << " " << "Derived1 : " << d.Derived1 << endl;

cout << " " << "Derived2 : " << d.Derived2 << endl;

}

};

class Friend2\_Class

{

Friend\_Class a;

public:

void accessMethod(){

cout << "Access from friend of Friend\_Class\n";

a.accessMethod();

}

};

int main()

{

cout << "BASE CLASS : " << endl;

Base base(13, 25, 10);

base.print();

cout << endl << endl;

cout << "DERIVED CLASS : " << endl;

Derived derived(11, 4);

derived.print();

cout <<endl<<endl;

cout << "FRIEND CLASS" << endl;

Friend\_Class fri;

fri.accessMethod();

cout << endl << endl;

cout << "FRIEND1\_CLASS" << endl;

Friend1\_Class fri1;

fri1.accessMethod();

cout <<endl<<endl;

cout << "FRIEND2\_CLASS" << endl;

Friend2\_Class fri2;

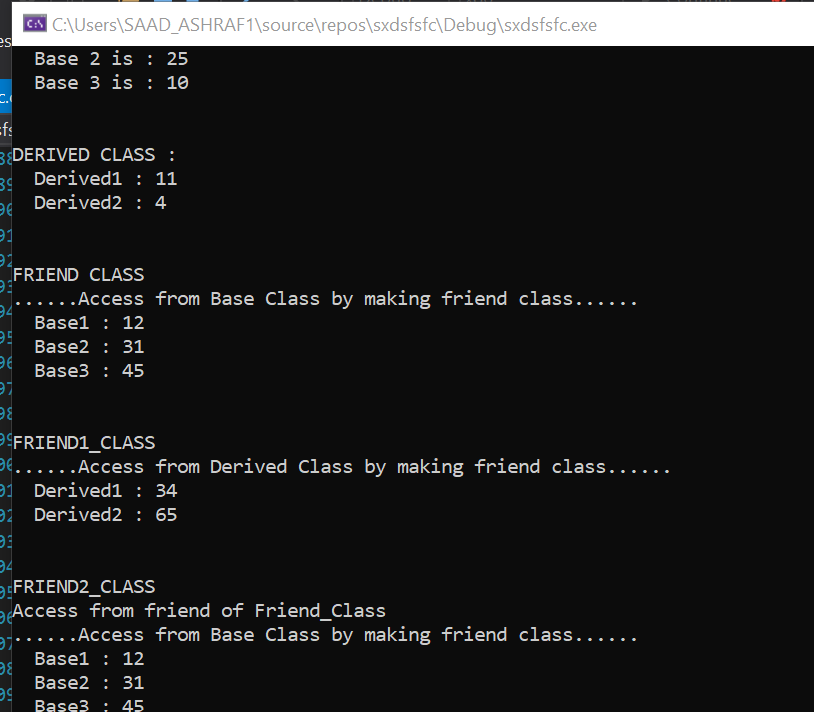
fri2.accessMethod();

cout << "\n";

system("pause");

return 0;

}



Question 5:

#include<iostream>

using namespace std;

class test {

private:

string x, y;

friend void Func(test&);

friend const void test2(test&);

public:

test(){

x = "Saad";

y = "ashraf";

}

};

void Func(test& t){

t.x = "abcd";

t.y = "efgh";

cout << " X is : " << t.x << endl;

cout << " Y is : " << t.y << endl;

}

const void test2(test& t) {

cout << "Value of X : " << t.x << endl;

cout << "Value of Y : " << t.y << endl;

}

int main()

{

test var;

test2(var);

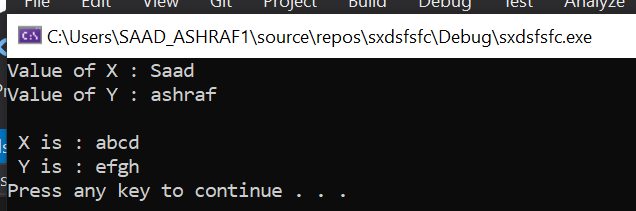
cout <<endl;

Func(var);

system("pause");

return 0;

}



Question 6:

#include<iostream>

#include<string>

using namespace std;

class publication

{

private:

string title;

float price;

public:

virtual void getData() {

cout << "\nEnter Title of the Publication :";

cin >> title;

cout << "\nEnter the price of Publication :";

cin >> price;

}

virtual void putData()

{

cout << "Title :" << title << endl;

cout << "Price :" << price << endl;

}

};

class book :public publication

{

int page\_count;

public:

void getData() {

cout << "\nEnter the page count of Book :";

cin >> page\_count;

}

void putData() {

cout << "Page count of Book is :" << page\_count << endl;

}

};

class tape :public publication {

float ptim;

public:

void getData() {

cout << "\nEnter playing time in minutes : ";

cin >> ptim;

}

void putData() {

cout << "Playing time in minutes :" << ptim << endl;

}

};

int main()

{

int step;

publication\* a[100];

for (int i = 0; i < 100; i++) {

cin >> step;

if (step == 1) {

a[i] = new publication;

a[i]->publication::getData();

a[i]->publication::putData();

a[i] = new book;

a[i]->getData();

a[i]->putData();

a[i] = new tape;

a[i]->getData();

a[i]->putData();

cout << "Press 1 to continue or any other to end program" << endl;

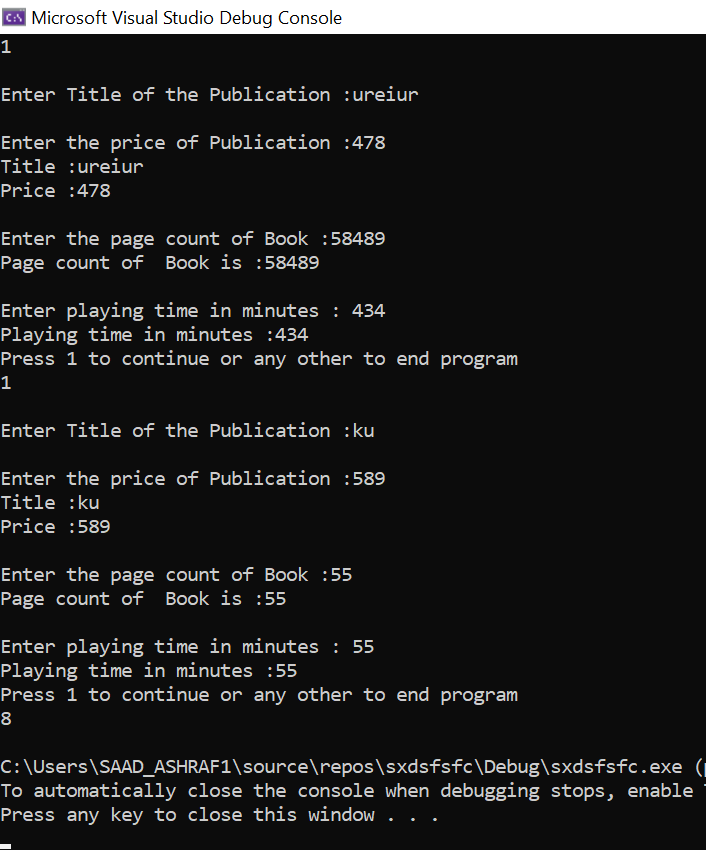
}

else

break;

}

}



Question 8:

#include<iostream>

using namespace std;

class polygon{

public:

float length, width;

virtual float area() = 0;

virtual float perimeter() = 0;

virtual void display() = 0;

void get\_length(float length){

this->length = length;

}

void get\_width(float width){

this->width = width;

}

float set\_length(){

return length;

}

float set\_width(){

return width;

}

};

class square :public polygon{

public:

square() {}

float area(){

return length \* width;

}

float perimeter(){

return 4 \* length;

}

void display(){

cout << "Area of the sqaure : " << area() << endl;

cout << "Perimeter of the square : " << perimeter() << endl;

}

};

class rectangle :public polygon{

public:

rectangle() {}

rectangle(float lenth){

this->length = length;

}

float area(){

return (2 \* (length \* width));

}

float perimeter(){

return (2 \* (length \* width));

}

void display(){

cout << "Area of the rectangle : " << area() << endl;

cout << "Perimeter of the rectangle : " << perimeter() << endl;

}

};

class triangle :public polygon{

public:

triangle() {}

triangle(float length, float width){

this->length = length;

this->width = width;

}

float area(){

return ((length \* width) / 2);

}

float perimeter(){

return (length + width);

}

void display(){

cout << "Area of the triangle : " << area() << endl;

cout << "Perimeter of the Triangle : " << perimeter() << endl;

}

};

int main()

{

float l1, l2, l3, w1, w2, w3;

cout << "Enter Length of square : ";

cin >> l1;

cout << "Enter width of square : ";

cin >> w1;

cout << "Enter Length of rectangle : ";

cin >> l2;

cout << "Enter width of rectangle : ";

cin >> w2;

cout << "Enter Length of traingle : ";

cin >> l3;

cout << "Enter width of traingle : ";

cin >> w3;

int i = 0;

polygon\* p[3];

p[0] = new square;

p[0]->get\_length(l1);

p[0]->get\_width(w1);

p[1] = new rectangle;

p[1]->get\_length(l2);

p[1]->get\_width(w2);

p[2] = new triangle;

p[2]->get\_length(l3);

p[2]->get\_width(w3);

while (i < 3){

p[i]->display();

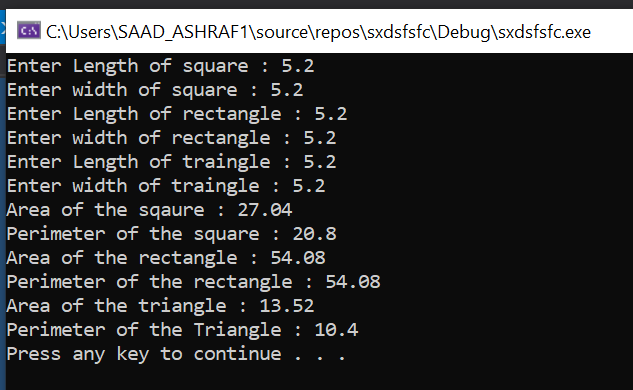
i++;

}

system("pause");

return 0;

}



Question 9:

#include<iostream>

using namespace std;

class Polygon

{

protected:

int width, height;

public:

void set\_values(int a, int b)

{

width = a; height = b;

}

virtual int area() = 0;

};

class Rectangle : public Polygon {

public:

int area()

{

return width \* height;

}

};

class Triangle : public Polygon {

public:

int area()

{

return width \* height / 2;

}

};

int main() {

Rectangle rect;

Triangle trgl;

Polygon\* ppoly1 = &rect;

Polygon\* ppoly2 = &trgl;

ppoly1->set\_values(4, 5);

ppoly2->set\_values(4, 5);

int m = ppoly2->area();

cout << "Value of area: " << m << endl;

system("pause");

return 0;

}

