The final is comprehensive. It has some T/F, most are questions to read code or write code.

* The questions for modules 1-5 are similar to the midterms.

Topics: 2D arrays, binary search, insertion sort, dynamic memory allocation, C strings, C++ string class, structs, classes and objects

* The new material is module 7, so here are the review questions on modules 5, 6, 7

1. Given this class alpha:

class alpha

{

public:

alpha(int input=1) {num = input;}

alpha operator++() {++num; return num;}

int operator=(int x) {num = x+1; return num;}

friend ostream & operator<< (ostream & o, alpha & a)

{o << a.num - 1; return o;}

private:

int num;

};

Show what value each alpha object has after each line of code, or write error if there is an error

void main()

{

alpha A;

alpha B = A++;

A = 10;

cout << A;

A = B;

}

2. Given the following classes:

class base

{

public:

base(int in = 0) {ptr = new int; \*ptr = in;}

~base() {delete ptr;}

void show() {cout << \*ptr << endl;}

virtual void F(int in) {\*ptr = in;}

protected:

int \* ptr;

};

class derived : public base

{

public:

derived(int in) {num = in;}

void show() {cout << num << endl;}

void F(int in) {num = in;}

private:

int num;

};

void main()

{

base B(5);

B.show();

B.derived::show();

B.F(15);

B.show();

derived D(10);

D.show();

D.F(8);

D.show();

D.base::show();

base \*p = new derived(30);

p->F(20);

p->show();

delete p;

p = new base(40);

p->F(50)

p->show();

delete p;

}

Given the code from main above,

a. If there is any error, explain what the error is

b. Show what is printed to screen

c. For the pointer p:

What type of object is p first pointing to?

How many constructors are run to create this object?

How many destructors run at the first delete p?

3. You have the following 2 classes, which are partially written: a point base class and a TDpoint (for 3D point) derived class.

Add the function prototypes/definitions to the 2 classes for main to work as written. Don’t add extra functions that will not be needed.

class point

{

point(int x = 1, int y = 1) {this->x = x; this->y = y;}

void show() const { cout << x << “, ” << y; }

int x, y; // x, y coordinates

};

class TDpoint : public point

{

int z; // z coordinate

};

void main()

{

TDpoint A(1,2,3);

A.show; // print all 3 data fields

TDpoint C = B \* 3; // multiply x, y, z by 3

point D = A // D has A's x, y values

display(D); // display x,y of D

display(C); // display x,y,z of C

}