***Câu1:***

a) Thế nào là lớp bản mẫu? Điều kiện để sử dụng một lớp dữ liệu cụ thể với một lớp bản mẫu là gì?

b) Khi nào hàm tĩnh được sử dụng, cho ví dụ?

- Lớp bản mẫu: là lớp thao tác với tham số kiểu bản mẫu và tất cả các hàm thành viên của nó đều là hàm bản mẫu.

- ĐK: Tập hợp các lớp chứa bản mẫu(là các lớp được sinh ra để quản lí đối tượng của các lớp khác, thường là các lớp cấu trúc, DL điển hình) và các hàm bản mẫu hay dùng tạo thành thư viện bản mẫu chuẩn STL <Standard Temple Library>.

Hàm tĩnh :

Có tác dụng giúp bạn có thể gọi đến hàm đó mà không cần phải khởi tạo object gì cả. Người ta có thể dùng static function khi muốn truy cập vào 1 biến static và “private”. Tức là nếu một biến là private, bạn không thể truy cập nó từ bên ngoài class. Thay vì sử dụng 1 member function bình thường thì phải tạo 1 object, thì có thẻ dùng static function để gọi trực tiếp mà không cần tạo.

***Câu 2:***

a) Thế nào là hàm toán tử gọi hàm, cho ví dụ?

b) Hãy cài đặt hàm toán tử tăng dạng hậu tố cho lớp ma trận các số nguyên kích thước 2 x 2.

Hàm toán tử <operator method>: là hàm mà việc gọi hàm có thể dùng dấu toán tử thay vì tên hàm <đầy đủ>.

Toán tử gọi hàm () trong C++ có thể được nạp chồng cho các đối tượng của kiểu lớp. Khi bạn nạp chồng (), bạn đang không tạo một cách mới để gọi một hàm. Đúng hơn là, bạn đang tạo một hàm toán tử mà có thể được truyền số tham số tùy ý.

VD: hàm toán tử nhập xuất, toán hạng(+, -, \*, /), so sánh(>, <, ==, !=,...), [ ](chỉ số), ( ) (gọi hàm).

A1(2, 8); // Có thể gọi hàm tên là A1 hoặc gọi tới hàm toán tử gọi hàm của đối tượng A1

#include <iostream>

using namespace std;

int operator (int a, int b){

return (a\*b);

}

double operator (double a, double b){

return (a/b);

}

int main (){

int x=5,y=2;

double n=5.0,m=2.0;

cout << operator (x,y) << '\n';

cout << operator (n,m) << '\n';

return 0;

}

***Câu 3:***

- Trình bày sự hiểu biết của em về thành phần static trong Lập trình HĐT;

- Hãy giải thích và sửa lại những chỗ sai trong khai báo sau:

class Time {

public:

void ~Time(int); //hủy tử

private:

int h = 0; m = 0; n = 0;

static int d = 0;

};

Static :

-Từ khóa static có thể được sử dụng để khai báo biến, dù đặt bên trong hay bên ngoài những khối lệnh, vẫn khiến cho biến đó có hiệu lực tương tự như biến toàn cục.

-Từ khóa static còn có thể được sử dụng trong việc định nghĩa hàm, các thuộc tính và phương thức trong một lớp (class).

-Biến static được tạo ra bên trong một khối lệnh có khả năng lưu giữ giá trị của nó cho dù chương trình đã chạy ra bên ngoài khối lệnh chứa nó.

-Biến static chỉ cần được khai báo một lần duy nhất, và tiếp tục được duy trì sự tồn tại xuyên suốt cho đến khi chương trình kết thúc.

***Câu4:***

- Trình bày sự hiểubiết của bạn về bản mẫu hàm.

- Xây dựng bản mẫu hàm tMax để có thể tìm được vị trí phần tử lớn nhất từ phần tử thứ N đến phần tử thứ M trong mảng A.

- Hàm bản mẫu là một kĩ thuật của LT HDT, cho phép tham số hóa kiểu của tham số cuả hàm. Kiểu của tham số là một tham số. Có thể coi kỹ thuật này là một sự trừu tượng (tổng quát hóa) về kiểu: 1 thao tác được thực hiện với các giá trị khác nhau của các kiểu dữ liệu khác nhau, khác với trừu tượng về giá trị. Một thao tác được thực hiện với các giá trị khác nhau của cùng một kiểu.

- Một hàm bản mẫu là 1 mẫu <template> để sinh ra các hàm cụ thể được sử dụng trong chương trình – do trình biên dịch sinh ra. Mã nguồn chương trình chỉ có một hàm chung \_\_ mã thực hiện thì được mở rộng thành nhiều hàm khác nhau để thực hiện với các kiểu DL khác nhau.

- . Có 2 loại “template” cơ bản:

-Function template: là một khuôn mẫu hàm, cho phép định nghĩa các hàm tổng quát thao tác cho nhiều kiểu dữ liệu.

-Class template: là một khuôn mẫu lớp, cho phép định nghĩa các lớp tổng quát cho nhiều kiểu dữ liệu.

***Câu 5:***

- Trình bày sự hiểu biết của bạn về quá tải toán tử. Nêu ví dụ về toán tử ++, --

- Bạn hãy sửa đoạn chương trình sau nếu có lỗi và cho biết kết quả hiển thị sau khi thực hiện hàm main().

class A {

private: int x,y;

public:

A(int x1, int y1) { x=x1; y=y1; } //cấu tử

void operator+=(A a) { x+=a.x; y+=a.y; }

void operator=+(A a) { x+=a.x; y+=a.y; }

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

};

Int main() {

A a1(5,5);

A a2(6,6);

a2+=a1;

a1=+a2;

a2.In(); a1.In();

}

Quá tải toán tử :

+, -, \*, /… về bản chất là các hàm với tham số là các toán hạng. Trong OOP cho phép định nghĩa lại (quá tải) các toán tử có kiểu dữ liệu khác so với dữ liệu ban đầu. Hầu hết các toán tử đều có thể quá tải trừ 1 số toán tử như toán tử điều kiện 3 ngôi, phạm vi ::, toán tử truy cập vào thành phần của lớp.

**Điểm sai 1: void operator-=……: Khai báo sai hàm**

**Sửa: A operator-=….**

**Điểm sai 2: Chưa có toán tử +=**

**Sửa: thêm hàm sau vào class A**

**Friend A operator +=(A a, A b){ a.x+=b.x; a.y+=b.y; return a; }**

**Kết quả sau khi chạy: 15, 15**

**-20, -20**

**Or sửa điểm sai 2: bỏ**

**Kết quả : 10, 10**

**-20, -20**

***Câu6:***

a) Thế nào là toán tử gọi hàm, cho ví dụ?

b) Sửa lỗi trong đoạn chương trình sau:

class B {

int id;

public:

static int getID () { return id;};

static int num = 0;

};

1. Giống hệt câu 3 phần a đề 2

using namespace std;

class B {

int id;

public:

int getID () { return id;};

static int num ;

};

***Câu 7:***

a) Trình bày các hiểu biết của em về lớp trừu tượng, cho ví dụ?

b) Khai báo và cài đặt hàm bản mẫu sắp xếp một mảng các đối tượng.

Lớp trừu tượng (thường gọi tắt là ABC) là để cung cấp một lớp cơ sở thích hợp để từ đó các lớp khác có thể kế thừa. Các lớp trừu tượng không thể được sử dụng để khởi tạo các đối tượng và chỉ phục vụ như là một Interface. Nỗ lực để khởi tạo một đối tượng của một lớp trừu tượng trong C++ sẽ tạo ra một lỗi biên dịch (compilation error).

Vì thế, nếu một lớp phụ của một ABC cần được khởi tạo, nó phải triển khai các hàm virtual, nghĩa là nó hỗ trợ Interface được khai báo bởi ABC. Thất bại trong việc ghi đè một hàm pure virtual trong một lớp kế thừa, thì việc nỗ lực để khởi tạo các đối tượng của lớp đó, là một lỗi biên dịch.

Các lớp, mà có thể được sử dụng để khởi tạo đối tượng, được gọi là Concrete Class trong C++.

class Hinh {

int chieurong, chieucao;

public:

virtual int tinhDienTich() = 0;

void setChieuRong(int rong){

chieurong = rong;

}

void setChieuCao(int cao){

chieucao = cao; }

};

class HinhChuNhat: public Hinh{

public:

int tinhDienTich(){

return (chieurong \* chieucao);

}

};

class TamGiac: public Hinh{

public:

int tinhDienTich(){

return (chieurong \* chieucao)/2;

} };

***Câu8:***

a) Trình bày các hiểu biết của em về hàm ảo thực sự?

b) Khai báo, cài đặt và sử dụng một hàm tĩnh tính độ dài của một đoạn thẳng biết tọa độ điểm đầu và điểm cuối.