

Lập trình hướng đối tượng (OOP)

Câu 1. Lập trình hướng đối tượng là:

- a. Lập trình hướng đối tượng là phương pháp đặt trọng tâm vào các đối tượng, nó không cho phép dữ liệu chuyển động một cách tự do trong hệ thống; dữ liệu được gắn với các hàm thành phần.
- b. Lập trình hướng đối tượng là phương pháp lập trình cơ bản gắn với mã máy.
- c. Lập trình hướng đối tượng là phương pháp mới của lập trình máy tính, chia chương trình thành các hàm; quan tâm đến chức năng của hệ thống.
- d. Lập trình hướng đối tượng là phương pháp đặt trọng tâm vào các chức năng, cấu trúc chương trình được xây dựng theo cách tiếp cận hướng chức năng.

Câu 2. Đặc điểm cơ bản của lập trình hướng đối tượng thể hiện ở:

- a. Tính đóng gói, tính kế thừa, tính đa hình, tính trừu tượng.
- b. Tính đóng gói, tính kế thừa, tính đa hình, tính đặc biệt hóa.
- c. Tính chia nhỏ, tính kế thừa.
- d. Tính đóng gói, tính trừu tượng.

Câu 3. OOP là viết tắt của:

- a. Object Oriented Programming.
- b. Object Open Programming.
- c. Open Object Programming
- d. Object Oriented Proccessing.

Câu 4. Hãy chọn câu đúng:

- a. Ngôn ngữ lập trình pascal, C là ngôn ngữ lập trình cấu trúc.
- b. Ngôn ngữ lập trình pascal, C, C++ là ngôn ngữ lập trình cấu trúc.
- c. Ngôn ngữ lập trình C, C++ là ngôn ngữ lập trình hướng đối tượng.
- d. Ngôn ngữ lập trình C++, Java là ngôn ngữ lập trình cấu trúc.

Câu 5. Chọn câu sai:

- a. Ngôn ngữ Pascal là ngôn ngữ lập trình hướng đối tượng.
- b. Ngôn ngữ C++, Java là ngôn ngữ lập trình hướng đối tượng.
- c. C, Pascal là ngôn ngữ lập trình cấu trúc.
- d. C++ là ngôn ngữ lập trình cấu trúc.

Câu 6. Tính bao gói là:

- a. Cơ chế ràng buộc dữ liệu và thao tác trên dữ liệu đó thành một thể thống nhất, tránh được các tác động bất ngờ từ bên ngoài. Thể thống nhất này gọi là đối tượng.
- b. Cơ chế chia chương trình thành các hàm và thủ tục thực hiện các chức năng riêng rẽ.
- c. Cơ chế không cho phép các thành phần khác truy cập đến bên trong nó.
- d. Cơ chế cho thấy một hàm có thể có nhiều thể hiện khác nhau ở từng thời điểm.

Câu 7. Tính kế thừa là:

- a. Khả năng xây dựng các lớp mới từ các lớp cũ, lớp mới được gọi là lớp dẫn xuất, lớp cũ được gọi là lớp cơ sở.
- b. Khả năng sử dụng lại các hàm đã xây dựng.
- c. Khả năng sử dụng lại các kiểu dữ liệu đã xây dựng.
- d. Tất cả đều đúng.

Câu 8. Sự đóng gói là:

- a. Encapsulation
- b. Inheritance.
- c. Polymorphism.
- d. Abstraction.

Câu 9. Sự trừu tượng là:

- a. Encapsulation
- b. Inheritance.
- c. Polymorphism.
- d. Abstraction.

Câu 10. Sự thừa kế là:

- a. Encapsulation
- b. Inheritance.
- c. Polymorphism.
- d. Abstraction.

Câu 11. Tính đa hình là:

- a. Encapsulation
- b. Inheritance.
- c. Polymorphism.
- d. Abstraction.

Câu 12. Trong kế thừa. Lớp mới có thuật ngữ tiếng Anh là:

- a. Derived Class.
- b. Base Class.
- c. Inheritance Class.
- d. Object Class.

Câu 13. Trong kế thừa. Lớp cha có thuật ngữ tiếng Anh là:

- a. Derived Class.
- b. Base Class.
- c. Inheritance Class.
- d. Object Class.

Câu 14. Tính đa hình là:

- a. Khả năng một thông điệp có thể thay đổi cách thể hiện của nó theo lớp cụ thể của đối tượng được nhận thông điệp.
- b. Khả năng một thông điệp có thể được truyền lại cho lớp con của nó.
- c. Khả năng một hàm, thủ tục có thể được kế thừa lại.
- d. Khả năng một hàm, thủ tục được sử dụng lại.

Câu 15. Lớp đối tượng là:

- a. Một thiết kế hay mẫu cho các đối tượng cùng kiểu.
- b. Một thể hiện cụ thể cho các đối tượng.
- c. Tập các phần tử cùng loại.
- d. Tập các giá trị cùng loại.

Câu 16. Sau khi khai báo và xây dựng thành công lớp đối tượng Sinh viên. Khi đó lớp đối tượng Sinh viên còn được gọi là:

- a. Đối tượng.
- b. Kiểu dữ liệu trừu tượng.
- c. Kiểu dữ liệu cơ bản.
- d. Lớp đối tượng cơ sở.

Câu 17. Trong các phương án sau, phương án nào mô tả đối tượng:

- a. Tất cả đều đúng.
- b. Quả cam.
- c. Xe đạp.

d. Máy tính.

e. Đồng hồ.

Câu 18. Muốn lập trình hướng đối tượng, bạn cần phải phân tích chương trình, bài toán thành các:

a. Hàm, thủ tục

b. Các module

c. Các đối tượng từ đó xây dựng các lớp đối tượng tương ứng.

d. Các thông điệp

Câu 19. Trong phương án sau, phương án mô tả tính đa hình là:

a. Các lớp Điểm, Hình tròn, Hình vuông, Hình chữ nhật... đều có phương thức Vẽ.

b. Lớp hình tròn kế thừa lớp điểm.

c. Lớp hình vuông kế thừa lớp hình chữ nhật.

d. Lớp Điểm, Hình tròn cùng có hàm tạo, hàm hủy.

Câu 20. Phương pháp lập trình tuần tự là:

a. Phương pháp lập trình với cách liệt kê các lệnh kế tiếp.

b. Phương pháp lập trình với việc cấu trúc hóa dữ liệu và cấu trúc hóa chương trình để tránh các lệnh nhảy.

c. Phương pháp lập trình được cấu trúc nghiêm ngặt với cấu trúc dạng module.

d. Phương pháp xây dựng chương trình ứng dụng theo quan điểm dựa trên các cấu trúc dữ liệu trừu tượng, các thể hiện cụ thể của cấu trúc và quan hệ giữa chúng..

Câu 21. Phương pháp lập trình cấu trúc là:

a. Phương pháp lập trình với cách liệt kê các lệnh kế tiếp.

b. Phương pháp lập trình với việc cấu trúc hóa dữ liệu và cấu trúc hóa chương trình để tránh các lệnh nhảy.

c. Phương pháp lập trình được cấu trúc nghiêm ngặt với cấu trúc dạng module.

d. Phương pháp xây dựng chương trình ứng dụng theo quan điểm dựa trên các cấu trúc dữ liệu trừu tượng, các thể hiện cụ thể của cấu trúc và quan hệ giữa chúng..

Câu 22. Khi khai báo và xây dựng thành công lớp đối tượng, để truy cập vào thành phần của lớp ta phải:

a. Truy cập thông qua tên lớp hay tên đối tượng của lớp.

b. Không thể truy cập được.

c. Chỉ có thể truy cập thông qua tên lớp.

b. Chỉ có thể truy cập thông qua tên đối tượng của lớp.

Câu 23. Phương pháp lập trình module là:

a. Phương pháp lập trình với cách liệt kê các lệnh kế tiếp.

b. Phương pháp lập trình với việc cấu trúc hóa dữ liệu và cấu trúc hóa chương trình để tránh các lệnh nhảy.

c. Phương pháp lập trình được cấu trúc nghiêm ngặt với cấu trúc dạng module.

d. Phương pháp xây dựng chương trình ứng dụng theo quan điểm dựa trên các cấu trúc dữ liệu trừu tượng, các thể hiện cụ thể của cấu trúc và quan hệ giữa chúng..

Câu 24. Trừu tượng hóa là:

a. Phương pháp chỉ quan tâm đến những chi tiết cần thiết (chi tiết chính) và bỏ qua những chi tiết không cần thiết

b. Phương pháp quan tâm đến mọi chi tiết của đối tượng

c. Phương pháp thay thế những chi tiết chính bằng những chi tiết tương tự.

d. Không có phương án chính xác

Câu 25. Đối tượng là:

a. Một thực thể cụ thể trong thế giới thực

b. Một mẫu hay một thiết kế cho mọi lớp đối tượng.

c. Một vật chất trong thế giới thực.

d. Một lớp vật chất trong thế giới thực.

Câu 26. Khi khai báo và xây dựng một lớp ta cần phải xác định rõ thành phần:

a. Thuộc tính (dữ liệu) và phương thức (hành vi) của lớp.

b. Dữ liệu và đối tượng của lớp.

c. Khái niệm và đối tượng của lớp.

d. Vô số thành phần.

Câu 27. Đoạn chương trình sau khi biên dịch pháp sinh ra lỗi không khai báo biến x tại dòng cin;

```
class A
```

```
{ private:
```

```
int x;
```

```
public:
```

```
void nhap();
```

```
};
```

```
void nhap();
```

```
{
```

```
cout<< "\n Nhap x; cin>>x;
```

```
}  
void main() {}  
Cần sửa là:
```

- a. Sửa dòng void nhap(); thứ hai thành void A::nhap();
- b. Khai báo thêm biến x trong hàm nhap;
- c. Chương trình bị lỗi không chạy được.
- d. Tất cả đều đúng.

Câu 28. Chọn câu đúng:

- a. Tại chương trình chính chỉ có thể truy cập đến thành phần public của lớp.
- b. Tại chương trình chính chỉ có thể truy cập đến thành phần private của lớp.
- c. Tại chương trình chính có thể truy cập đến bất kì thành phần nào của lớp.
- d. Tại chương trình chính không thể truy cập đến thành phần nào của lớp.

Câu 29. Cho đoạn chương trình sau:

```
class A {  
private:  
int x;  
public:  
int y;  
};  
void main() {  
A obj1; obj1.x=10;  
obj2.y=8;  
}
```

Đoạn chương trình bị lỗi tại:

- a. Lỗi tại dòng obj1.x=10;
- b. Lỗi tại dòng obj1.y=8;
- c. Lỗi tại dòng A obj1;
- d. Lỗi tại dòng int x;

Câu 30. Cho đoạn chương trình sau:

```
class A {  
private:  
int x;  
public:  
int y;  
};  
void main() {  
A obj1;  
obj1.x=10;
```

```
obj2.y=8;  
}
```

Đoạn chương trình bị lỗi tại obj1.x=10. Nguyên nhân của lỗi này là:

- a. Không thể truy cập vào thành phần private của lớp.
- b. Chưa khai báo x.
- c. Không xác định được giá trị x.
- d. Phải gọi thông qua tên lớp không được gọi thông qua tên đối tượng.

Câu 31. Khi khai báo lớp trong các ngôn ngữ lập trình hướng đối tượng phải sử dụng từ khóa:

- a. Object.
- b. Record.
- c. File.
- d. class.

Câu 32. Thành phần private của lớp là thành phần:

- a. Không cho phép truy xuất từ bên ngoài của lớp chỉ có các phương thức bên trong lớp mới có thể truy xuất được.
- b. Không cho phép truy xuất từ bên ngoài của lớp nhưng cho phép lớp kế thừa truy xuất tới.
- c. Cho phép truy xuất từ bên ngoài lớp.
- d. Cho phép truy xuất từ bên ngoài lớp và cho phép kế thừa.

Câu 33. Thành phần protected của lớp là thành phần:

- a. Không cho phép truy xuất từ bên ngoài của lớp chỉ có các phương thức bên trong lớp mới có thể truy xuất được.
- b. Không cho phép truy xuất từ bên ngoài của lớp nhưng cho phép lớp kế thừa truy xuất tới.
- c. Cho phép truy xuất từ bên ngoài lớp.
- d. Cho phép truy xuất từ bên ngoài lớp và cho phép kế thừa.

Câu 34. Thành phần public của lớp là thành phần:

- a. Không cho phép truy xuất từ bên ngoài của lớp chỉ có các phương thức bên trong lớp mới có thể truy xuất được.
- b. Không cho phép truy xuất từ bên ngoài của lớp nhưng cho phép lớp kế thừa truy xuất tới.
- c. Cho phép truy xuất từ bên ngoài lớp.
- d. Cho phép truy xuất từ bên trong và ngoài lớp và cho phép kế thừa.

Câu 35. Hàm thành viên (phương thức) của lớp:

- a. Tất cả các hàm(hàm trả về giá trị và không trả về giá trị) được khai báo bên trong lớp, xây dựng bên trong hay bên ngoài lớp thể hiện các hành vi của đối tượng.
- b. Tất cả các hàm(hàm và thủ tục) được sử dụng trong lớp.
- c. Tất cả những hàm (hàm và thủ tục) trong chương trình có lớp.
- d. Tất cả những hàm(hàm và thủ tục) được khai báo và xây dựng bên trong lớp mô tả các dữ liệu của đối tượng.

Câu 36. Trong một chương trình có thể xây dựng tối đa bao nhiêu lớp

- a. Vô số tùy theo bộ nhớ.
- b. 1 lớp duy nhất
- c. 10 lớp
- d. 3 lớp

Câu 37. Hàm thành viên của lớp khác hàm thông thường là:

- a. Hàm thành viên của lớp phải được khai báo bên trong lớp và được gọi nhờ tên đối tượng hay tên lớp còn hàm thông thường thì không.
- b. Hàm thành viên của lớp và hàm thông thường không có gì khác nhau.
- c. Hàm thành viên của lớp thì phải được khai báo và xây dựng bên trong lớp còn hàm thông thường thì không.
- d. Hàm thành viên của lớp thì phải khai báo bên trong lớp với từ khóa friend và xây dựng bên ngoài lớp.

Câu 38. Thuộc tính của lớp là:

- a. Là dữ liệu trình bày các đặc điểm của một đối tượng.
- b. Liên quan tới những thứ mà đối tượng có thể làm. Một phương thức đáp ứng một chức năng tác động lên dữ liệu của đối tượng.
- c. Là những chức năng của đối tượng.
- d. Tất cả đều sai.

Câu 39. Phương thức là:

- a. Là dữ liệu trình bày các đặc điểm của một đối tượng.
- b. Liên quan tới những thứ mà đối tượng có thể làm. Một phương thức đáp ứng một chức năng tác động lên dữ liệu của đối tượng.
- c. Là những chức năng của đối tượng.
- d. Tất cả đều đúng.

Câu 40. Cho lớp người hãy xác định đâu là các thuộc tính của lớp người:

- a. Chân, Tay, Mắt, Mũi, Tên, Ngày sinh.
- b. Ăn, Uống, Chân, Tay.
- c. Hát, học, vui, cười.
- d. Tất cả đều sai.

Câu 41. Người ta cần quản lý thông tin sinh viên trên máy tính, Hãy cho biết các thuộc tính của lớp sinh viên là:

- a. Họ và tên, ngày sinh, giới tính, địa chỉ, số cmt, quê quán
- b. Họ và tên, ngày sinh, giới tính, địa chỉ, số cmt, quê quán, nhóm máu, màu mắt, màu da, cân nặng.
- c. Họ và tên, ngày sinh, giới tính, địa chỉ, số cmt, quê quán, lớp học, khóa học, khoa quản lý.

Câu 42. Cho lớp Điểm trong hệ tọa độ xOy. Các phương thức có thể có của lớp Điểm là:

- a. Dịch chuyển, Thiết lập toạ độ.
- b. Tung độ, hoành độ.
- c. Tung độ, hoành độ, cao độ.
- d. Tung độ, cao độ.

Câu 43. Lập trình hướng đối tượng:

- a. Tất cả đều đúng.
- b. Nhấn mạnh trên dữ liệu hơn là thủ tục.
- c. Các chương trình được chia thành các đối tượng.
- d. Dữ liệu được che giấu và không thể được truy xuất từ các hàm bên ngoài.
- e. Dữ liệu hay các hàm mới có thể được thêm vào khi cần.

Câu 44. Lời gọi tới hàm thành viên của lớp là:

- a. Tên_đối_tượng.Tên_hàm_thành_viên.
- b. Tên_lớp.Tên_hàm_thành_viên.
- c. Tên_lớp:Tên_hàm_thành_viên.
- d. Không có phương án đúng.

Câu 45. Khi khai báo thành phần thuộc tính và phương thức của lớp, nếu không khai báo từ khóa private, public hay protected thì mặc định sẽ là:

- a. private
- b. public
- c. protected

d. Chương trình sẽ lỗi và yêu cầu phải khai báo một trong 3 từ khóa.

Câu 46. Đoạn chương trình sau:

```
class A
{
private:
int x;
public:
void nhap();
};
void main()
{
A obj1;
obj1.nhap();
}
```

Khi biên dịch báo lỗi do:

- a. Không định nghĩa nội dung của phương thức nhap.
- b. Không có lỗi.
- c. Không được phép truy cập đến phương thức nhap.
- d. Không khai báo biến x.

Câu 47. Trong lập trình hướng đối tượng khả năng các hàm có thể trùng tên nhau gọi là:

- a. Không được phép xây dựng các hàm trùng tên nhau trong cùng một chương trình.
- b. Sự chồng hàm.
- c. Sự chồng hàm nhưng chỉ những hàm thành viên của lớp mới được phép trùng nhau.
- d. Sự chồng hàm nhưng chỉ các hàm thông thường mới được phép trùng nhau.

Câu 48. Phân tích mối quan hệ giữa lớp Điểm và lớp Hình tròn ta có thể xác định:

- a. Lớp Hình tròn kế thừa lớp Điểm.
- b. Lớp Điểm kế thừa lớp Hình tròn.
- c. Không có quan hệ gì.
- d. Lớp Hình tròn dẫn xuất ra lớp Điểm.

Câu 49. Hàm tạo trong ngôn ngữ C++:

- a. Có tên trùng với tên lớp.
- b. Có đối hoặc không có đối
- c. Tự động được gọi tới khi khai báo đối tượng của lớp.
- d. Xây dựng bên trong hoặc bên ngoài lớp
- e. Tất cả đều đúng

Câu 50. Hàm huỷ trong ngôn ngữ C++ có cú pháp:

- a. ~Tên_lớp{ //nội dung}.
- b. Destructor Tên_hàm {//nội dung}
- c. Tên_lớp{//nội dung}
- d. Done {//nội dung}

Câu 51. Cho đoạn chương trình sau

```
class A{  
private:  
int x,y;  
public:  
A(int x=0,int y=0);  
void xuat();  
~A();  
};  
A(int x=0,int y=0);
```

được gọi là:

- a. Hàm tạo
- b. Hàm huỷ
- c. Hàm bạn.
- d. Hàm thông thường.

Câu 52. Cho đoạn chương trình sau :

```
class A{  
private:  
int x,y;  
public:  
A(int x=0,int y=0);  
void xuat();  
~A();  
};  
~A();
```

được gọi là:

- a. Hàm tạo
- b. Hàm huỷ
- c. Hàm bạn.
- d. Hàm thông thường.
- e. Tất cả đều sai.

Câu 53. Cho đoạn chương trình

```
class A  
{  
private:
```

```
int x,y;  
};
```

Làm nhiệm vụ:

- a. Khai báo A là một lớp với hai thành phần thuộc tính là x, y kiểu số nguyên.
- b. Khai báo A là một đối tượng với hai thành phần thuộc tính x, y kiểu integer.
- c. Đoạn chương trình trên bị lỗi.
- d. Khai báo lớp A với hai thành phần thuộc tính x,y kiểu số nguyên cho phép truy cập từ bên ngoài lớp.

Câu 54. Cho khai báo sau:

```
class Diem {  
private:  
int x,y;  
public:  
voidNhap();  
voidXuat();  
};
```

x,y được gọi là:

- a. Thuộc tính của lớp Diem
- b. Phương thức của lớp Diem.
- c. Các hàm thành viên của lớp Diem
- d. Thành viên tĩnh của lớp Diem.

Câu 55. Cho khai báo sau:

```
class Diem {  
private:  
int x,y;  
public:  
voidNhap();  
voidXuat();  
};
```

Nhap, Xuat được gọi là:

- a. Phương thức của lớp Diem.
- b. Thuộc tính của lớp Diem.
- c. Các hàm bạn của lớp Diem
- d. Tất cả đều đúng.

Câu 56. Cho đoạn chương trình sau:

```
class Diem {  
float x,y;  
public:
```

...

};

class Doanthang{

Diem A,B;

public:

...

};

Lớp Doanthang được gọi là:

- a. Lớp bao của lớp Diem.
- b. Lớp cha của lớp Diem.
- c. Lớp con của lớp Diem.
- d. Lớp bạn của lớp Diem.

Câu 57. Lớp bao là lớp:

- a. Có thành phần thuộc tính là đối tượng của lớp khác.
- b. Kế thừa lớp khác.
- c. Dẫn xuất ra lớp khác.
- d. Lớp bạn của lớp khác.

Câu 58. Hàm tạo là:

- a. Hàm tạo là hàm thành viên của lớp dùng để khởi tạo bộ nhớ và giá trị ban đầu cho các thuộc tính trong lớp.
- b. Hàm tạo là hàm nằm bên ngoài lớp dùng để khởi tạo bộ nhớ cho đối tượng.
- c. Hàm tạo là hàm dùng để khởi tạo bộ nhớ cho đối tượng của lớp.
- d. Hàm tạo dùng để huỷ bộ nhớ cho đối tượng.

Câu 59. Hàm huỷ là:

- a. Hàm huỷ dùng để huỷ (giải phóng) bộ nhớ cho các thành phần thuộc tính bên trong lớp.
- b. Hàm huỷ là hàm dùng để khởi tạo giá trị ban đầu cho các thành phần thuộc tính bên trong lớp.
- c. Hàm huỷ là hàm dùng để giải phóng toàn bộ các biến của chương trình.
- d. Tất cả đều đúng.

Câu 60. Hàm tạo sao chép là:

- a. Tất cả đều đúng.
- b. Là hàm thành viên của lớp.
- c. Dùng để tạo một đối tượng theo đối tượng đã có.

d. Nếu không xây dựng hàm tạo sao chép chương trình sẽ tự sinh hàm tạo sao chép.

Câu 61. Cho đoạn chương trình sau:

```
class Sinh_vien
{
private:
int Masv;
char Hoten[40];
char Ngaysinh;
int gioitinh;
....
};
```

Để khai báo một mảng gồm 10 sinh viên ta có thể sử dụng câu lệnh khai báo:

- a. Sinh_vien ds[10];
- b. Sinh_vien ds(10);
- c. Không thể khai báo mảng các đối tượng
- d. Sinh_vien *ds[10];

Câu 62. Trong khai báo lớp Đa thức(DT) dưới đây bằng C++:

```
class DT {
private:
int m;//bac cua da thuc
float *a;//Cac he so cua da thuc
public:
};
```

Bắt buộc phải xây dựng:

- a. Hàm tạo.
- b. Hàm hủy.
- c. Hàm tạo sao chép
- d. Tất cả các hàm trên

Câu 63. Lời gọi hàm tạo:

- a. Gọi như hàm thành viên thông thường (Tên đối tượng.Tên_hàm) .
- b. Không cần phải gọi tới hàm tạo vì ngay khi khai báo đối tượng sẽ tự gọi tới hàm tạo.
- c. Gọi bằng cách: Tên lớp.Tên hàm tạo().
- d. Tất cả đều sai.

Câu 64. Ta khai báo lớp cơ sở ảo khi nào:

- a. Khi có sự trùng lặp lớp kế thừa trong đa kế thừa và kế thừa nhiều mức.
- b. Khi lớp có phương thức ảo thì bắt buộc phải khai báo là lớp cơ sở ảo.

- c. Khi có sự trùng tên giữa các phương thức của các lớp khác nhau.
- d. Tất cả đều sai.

Câu 65. Một người cần xây dựng lớp Thời gian (Timer) trong máy tính cần hiển thị thông tin như sau: giờ:phút:giây. Vậy các thuộc tính cần xây dựng cho lớp Timer là:

- a. Giờ, Phút, Giây.
- b. Giờ
- c. Phút
- d. Giây

Câu 66. Trong kế thừa, có thể:

- a. Kế thừa tất cả các phương thức thuộc tính khai báo trong phần protected, public và không kế thừa hàm tạo, hàm hủy.
- b. Kế thừa tất cả các phương thức thuộc tính khai báo trong phần protected, public bao gồm hàm tạo, hàm hủy.
- c. Kế thừa tất cả các phương thức thuộc tính khai báo trong phần protected, public, private bao gồm hàm tạo, hàm hủy.
- d. Kế thừa tất cả các phương thức thuộc tính khai báo trong phần protected, public không kế thừa hàm tạo, hàm hủy.

Câu 67. Các hàm tạo có thể có là:

- a. Hàm tạo không đối; Hàm tạo có đối; Hàm tạo sao chép; Hàm tạo bộ nhớ.
- b. Hàm tạo không đối; Hàm tạo có đối; Hàm tạo sao chép
- c. Hàm tạo không đối; Hàm tạo có đối.
- d. Hàm tạo không đối; Hàm tạo sao chép.

Câu 68. Cho khai báo sau:

```
int trituyetdoi(int a); //trị tuyệt đối số nguyên  
long trituyetdoi(long a); //tính trị tuyệt đối số nguyên dài  
double trituyetdoi(double a); //tính trị tuyệt đối số thực dài  
Khai báo trên thể hiện:
```

- a. Tính chất chồng hàm
- b. Tính chất nạp chồng của dữ liệu
- c. Tính chất đa hình
- d. Tính chất trừu tượng hóa

Câu 69. Lời gọi phương thức ảo là:

- a. Gọi như phương thức thông thường.

- b. Gọi kèm từ khoá virtual
- c. Không thể gọi phương thức ảo.
- d. Phải gọi thông qua con trỏ đối tượng.

Câu 70. Hàm hủy có:

- a. Một loại.
- b. Hai loại.
- c. Ba loại.
- d. Bốn loại.

Câu 71. Hãy chọn phát biểu sai:

- a. Có hai loại hàm hủy có đối và không đối.
- b. Có duy nhất một loại hàm hủy.
- c. Có hai loại hàm dựng có đối và không đối.
- d. Hàm hủy và hàm dựng đều không có tính chất kế thừa.

Câu 72. Trong khai báo dưới:

```
class Diem (1)
{(2)
private: int x,y;(3)
public: (4)
void Diem(int x,int y);(5)
void ~Diem();(6)
};(7)
```

Sai tại dòng:

- a. Dòng 6 và 5
- b. Dòng 3 và 5
- c. Dòng 2 và 3
- d. Dòng 3 và 6

Câu 73. Trong khai báo sau:

```
class Diem { (1)
private:
int x,y;(2)
public:(3)
.....
};(4)
class Hinhtron: public Diem (5)
{
private: r:real;(6)
```


};

Câu lệnh khai báo tại dòng số 5 cho biết:

- a. Khai báo lớp Hinhtron kế thừa public lớp Diem.
- b. Khai báo lớp Diem kế thừa lớp Hinhtron.
- c. Khai báo lớp Hinhtron là bạn lớp Diem.
- d. Khai báo lớp Hinhtron là lớp bao của lớp Diem.

Câu 74. Cho đoạn chương trình sau:

```
class A {  
private:  
int x,y;  
};
```

Cho lớp B kế thừa lớp A. Để truy cập vào thành phần x,y của lớp A ta chọn kiểu kế thừa:

- a. Không thể chọn loại kế thừa nào vì thành phần x,y thuộc phạm vi private không tham gia trong kế thừa.
- b. Chọn loại kế thừa private.
- c. Chọn loại kế thừa protected.
- d. Chọn loại kế thừa public.

Câu 75. Cho đoạn chương trình sau:

```
class A{  
private:  
int x,y;  
void Nhap();  
};
```

Cho lớp B kế thừa public lớp A, để lớp B có thể truy cập đến phương thức Nhap ta phải:

- a. Đưa phương thức Nhap của lớp A vào phạm vi public hoặc protected.
- b. Không cần phải thay đổi gì.
- c. Không thể truy cập được dù có thay đổi gì.
- d. Tất cả đều sai.

Câu 76. Cho đoạn chương trình sau:

```
class A{  
private:  
int x,y;  
protected:  
void Nhap();  
};
```

Lớp B kế thừa lớp A, để phương thức Nhap của lớp A trở thành thành phần private của lớp B ta chọn loại kế thừa là:

- a. Private.

- b. Public.
- c. Protected.
- d. Private hoặc Protected.

Câu 77. Cho đoạn chương trình sau:

```
class A {  
private:  
int x,y;  
public:  
void Nhap();  
};
```

Lớp B kế thừa lớp A, để phương thức Nhap của lớp A trở thành thành phần public của lớp B ta chọn loại kế thừa là:

- a. Private.
- b. Public.
- c. Protected.
- d. Private hoặc Protected

Câu 78. Các dạng kế thừa là:

- a. Private, Public, Protected
- b. Private, Public
- c. Private, Protected
- d. Protected, Public

Câu 79. Khi nạp chồng các hàm thì điều kiện khác nhau giữa các hàm sẽ là:

- a. Tên hàm phải khác nhau.
- b. Kiểu dữ liệu trả về của hàm (1)
- c. Kiểu dữ liệu của tham số truyền vào của hàm (2)
- d. Số lượng tham số truyền vào các hàm (3)
- e. Hoặc (1) hoặc (2) hoặc (3)

Câu 80. Trong một lớp có thể:

- a. Nhiều hàm dựng (tạo), các hàm dựng khác nhau về tham đối.
- b. Một hàm dựng.
- c. Hai hàm dựng.
- d. Tất cả đều sai.

Câu 81. Trong một lớp có thể:

- a. Duy nhất một hàm hủy.
- b. Chứa tối đa hai hàm hủy.
- c. Có thể chứa được ba hàm hủy.
- d. Có thể chứa vô số hàm hủy tùy theo bộ nhớ.

Câu 82. Tính chất của kế thừa dạng Private là:

- a. Thành phần protected, public của lớp cha tương ứng trở thành thành phần private của lớp con.
- b. Thành phần protected, public của lớp cha tương ứng trở thành thành phần public của lớp con.
- c. Thành phần protected, public của lớp cha tương ứng trở thành thành phần protected của lớp con.
- d. Tất cả đều sai.

Câu 83. Tính chất kế thừa dạng Protected là:

- a. Thành phần protected, public của lớp cha tương ứng trở thành thành phần protected của lớp con.
- b. Thành phần protected, public của lớp cha tương ứng trở thành thành phần public của lớp con.
- c. Thành phần protected, public của lớp cha tương ứng trở thành thành phần private của lớp con.
- d. Tất cả đều sai.

Câu 84. Tính chất kế thừa dạng Public là:

- a. Thành phần protected của lớp cha tương ứng trở thành thành phần protected của lớp con và thành phần public của lớp cha tương ứng trở thành public của lớp con.
- b. Thành phần protected, public của lớp cha tương ứng trở thành thành phần public của lớp con.
- c. Thành phần protected, public của lớp cha tương ứng trở thành thành phần private của lớp con.
- d. Tất cả đều sai.

Câu 85. Trong kế thừa có thể kế thừa tối đa:

- a. Vô số mức tùy theo bộ nhớ.
- b. Một mức.
- c. Hai mức.
- d. Ba mức.

Câu 86. Trong đa kế thừa có thể kế thừa tối đa:

- a. Vô số lớp tùy theo bộ nhớ.
- b. Hai lớp.
- c. Một lớp.
- d. Ba lớp.

Câu 87. Cho lớp A và lớp B, lớp A kế thừa lớp B trong hai lớp đều có phương thức Xuat. Nếu khai báo đối tượng obj thuộc lớp A khi gọi đến phương thức Xuat(obj.Xuat()) là gọi đến phương thức của lớp nào.

- a. Lớp A.
- b. Lớp B.
- c. Lỗi không thể xác định được.
- d. Gọi đến cả hai phương thức.

Câu 88. Cho lớp A, B, C trong đó lớp C kế thừa lớp A,B. Trong lớp A, B đều có phương thức Xuat, khi khai báo đối tượng obj thuộc lớp C thì nếu gọi đến phương thức Xuat (obj.Xuat()) thì phương thức Xuat của lớp nào sẽ được gọi?

- a. Lớp A.
- b. Lớp B.
- c. Lỗi không thể xác định được.
- d. Gọi đến cả hai phương thức.

Câu 89. Trong kế thừa nhiều mức có cho phép:

- a. Cho phép trùng tên cả phương thức lẫn thuộc tính.
- b. Không cho phép trùng tên phương thức và thuộc tính.
- c. Cho phép trùng tên phương thức còn không cho phép trùng tên thuộc tính.
- d. Cho phép trùng tên thuộc tính còn không cho phép trùng tên phương thức.

Câu 90. Cho đoạn chương trình sau:

```
class A{
...
public:
void Xuat();
};
class B:public A {
....
public:
void Xuat();
};
```

```
void main() {  
    B O;  
    Nếu gọi O.Xuat(); thì sẽ.
```

- a. Gọi đến phương thức Xuat thuộc lớp B.
- b. Lỗi.
- c. Gọi đến phương thức Xuat thuộc lớp A.
- d. Gọi đến cả hai phương thức.

Câu 91. Lớp cơ sở trừu tượng là:

- a. Là lớp làm cơ sở cho các lớp khác và không được dẫn xuất từ bất kì lớp nào.
- b. Là lớp cơ sở cho lớp khác.
- c. Là lớp không làm cơ sở cho bất kỳ lớp nào mà chỉ kế thừa các lớp khác.
- d. Là lớp được xây dựng đầu tiên trong chương trình.

Câu 92. Thành viên tĩnh của lớp là:

- a. Là thành viên của lớp được khai báo với từ khóa static ở trước.
- b. Là thành viên dùng chung cho tất cả các đối tượng của lớp, không của riêng đối tượng nào?
- c. Được cấp phát bộ nhớ ngay cả khi lớp chưa có đối tượng cụ thể nào.
- d. Tất cả đều đúng.

Câu 93. Hàm bạn là:

- a. Là hàm bạn của lớp được phép truy cập đến thành phần private của lớp.
- b. Khai báo với từ khóa virtual ở đầu.
- c. Được sử dụng như hàm thành viên của lớp.
- d. Tất cả đều đúng.

Câu 94. Khi xây dựng nạp chồng các hàm thì điều kiện khác nhau giữa các hàm là: kiểu dữ liệu trả về của hàm hoặc kiểu dữ liệu tham số truyền vào của hàm hoặc số lượng tham số truyền vào của hàm là khác nhau. Điều kiện này chỉ áp dụng khi:

- a. Các hàm cùng được xây dựng trong 1 chương trình (1).
- b. Các hàm này cùng được xây dựng trong một lớp (2).
- c. Các hàm này được xây dựng trong các lớp kế thừa.
- d. Các hàm này được xây dựng trong các lớp khác nhau.
- e. (1) hoặc (2)

Câu 95. Khai báo các phương thức trong lớp cơ sở trừu tượng phải:

- a. Là phương thức ảo thuần túy
- b. Với c++ có cú pháp : `virtual void[kiểu_dữ_liệu] tên_phương_thức(tham_đối)=0;`
- c. Các phương thức không được định nghĩa nội dung.
- d. Tất cả đều đúng.

Câu 96. Khai báo phương thức ảo:

- a. Giống khai báo phương thức thường theo sau là từ khoá `virtual`;
- b. Giống khai báo phương thức thường nhưng đứng đầu là từ khoá `virtual`;
- c. Giống khai báo phương thức thường nhưng phải được xây dựng bên trong lớp.
- d. Giống khai báo phương thức thường nhưng không cần phải xây dựng nội dung.

Câu 97. Cho biết cách khai báo phương thức ảo trong C++ nào sau đây là đúng:

- a. `void[kiểu_dữ_liệu] Tên_phương_thức ([các tham_đối]) Virtual;`
- b. `Virtual void[kiểu_dữ_liệu] Tên_phương_thức ([các tham_đối]);`
- c. `void[kiểu_dữ_liệu] Virtual Tên_phương_thức ([các tham_đối]);`
- d. `void[kiểu_dữ_liệu] Tên_phương_thức (Virtual [các tham_đối]);`

Câu 98. Cho đoạn chương trình sau:

```
class A {
private:
int x,y;
public:
void Xuat(); virtual;
};
```

Hãy nhận định phát biểu đúng:

- a. Xuat là phương thức ảo.
- b. Lỗi tại dòng `void Xuat(); virtual;`
- c. Xuat là một hàm tạo.
- d. Xuat là một hàm hủy.

Câu 99. Một lớp có thể có tối đa:

- a. Vô số phương thức ảo.
- b. Một phương thức ảo.
- c. Hai phương thức ảo.
- d. Không xác định được.

Câu 100. Cho đoạn chương trình sau:

```
class A {
```

```

int x,y;
public:
void Nhap();
void Xuat();
};
class B:public A{
private: float z;
};
void main()
{
A A1;B B1;
A1=B1;
B1=A1;
getch();
}

```

Hãy cho biết đoạn chương trình trên lỗi tại đâu?

- a. Tại câu lệnh B1=A1;
- b. Tại câu lệnh A1=B1
- c. Tại câu lệnh class B:public A.
- d. Không lỗi tại dòng nào cả?

Câu 101. Lập trình hướng đối tượng:

- a. Tất cả đều đúng.
- b. Nhấn mạnh trên dữ liệu hơn là thủ tục.
- c. Các chương trình được chia thành các đối tượng.
- d. Dữ liệu được che giấu và không thể được truy xuất từ các hàm bên ngoài.
- e. Dữ liệu hay các hàm mới có thể được thêm vào khi cần.

Câu 102. Cho biết các ví dụ sau ví dụ chứng tỏ tính kế thừa:

- a. Tất cả đều đúng.
- b. Xe đạp và Xe đạp đua.
- c. Hình bình hành và hình chữ nhật
- d. Điểm và Hình tròn.
- e. Giáo viên và Giáo viên giỏi

Câu 103. Lớp Bạn của một lớp là:

- a. Là lớp có thể truy cập đến thành phần private, protected và public của lớp mà nó làm bạn.
- b. Là lớp có thể truy cập đến thành phần protected và public của lớp mà nó làm bạn.

- c. Là lớp có thể truy cập đến thành phần public của lớp mà nó làm bạn.
- d. Tất cả đều sai.

Câu 104. Cho đoạn chương trình sau

```
class A;  
class B;  
class A{  
friend class B;  
...  
};  
class B {  
...  
};
```

Dòng lệnh friend class B; có nghĩa là:

- a. Khai báo lớp B là bạn lớp A.
- b. Khai báo lớp A là bạn lớp B
- c. Khai báo lớp B là bạn lớp A và ngược lại.
- d. Tất cả đều sai.

Câu 105. Cách khai báo kế thừa trong java sử dụng từ khóa:

- a. extends
- b. inheritance
- c. Sử dụng dấu : giống c++
- d. Khai báo giống c#

Câu 106. Cho đoạn chương trình sau:

```
class Trai_cay  
{  
private:  
char mauvo[5];  
char mauhat[3];  
char hinh dang[30];  
public:  
void Hienthi();  
};
```

Cần xây dựng lớp Trai_dua_hau. Quan hệ giữa lớp Trai_cay và lớp Trai_dua_hau là:

- a. Trai_cay là lớp cha của lớp Trai_dua_hau.
- b. Trai_dua_hau là lớp cha của lớp Trai_cay.
- c. Trai_cay và Trai_dua_hau là lớp bạn
- d. Trai_dua_hau là lớp bao của lớp Trai_cay.

Câu 107. Cho đoạn chương trình sau:

```
class Trai_cay
{
private:
char mauvo[5];
char mauhat[3];
char hinh dang[30];
public:
void Hienthi() const=0;
...
};
```

Lớp Trai_cay với khai báo bằng C++ như trên được gọi là lớp:

- a. Lớp cơ sở trừu tượng.
- b. Lớp cơ sở ảo.
- c. Lớp bao.
- d. Lớp dẫn xuất ảo.

Câu 108. Cho lớp tam giác tạo bởi ba điểm A, B, C. Quan hệ giữa lớp tam giác và lớp điểm là:

- a. Tam giác là lớp bao của lớp điểm.
- b. Tam giác là lớp con của lớp điểm.
- c. Tam giác là lớp cha của lớp điểm.
- d. Tam giác là lớp bạn của lớp điểm.

Câu 109. Cho lớp đường thẳng tạo bởi 2 điểm A, B. Quan hệ giữa lớp đường thẳng và lớp điểm là:

- a. Đường thẳng là lớp bao của lớp điểm.
- b. Đường thẳng là lớp con của lớp điểm.
- c. Đường thẳng là lớp cha của lớp điểm.
- d. Đường thẳng là lớp bạn của lớp điểm.

Câu 110. Cần xây dựng lớp đường thẳng $y=ax+b$. Thuộc tính của lớp đường thẳng này được xác định là:

- a. Các hệ số a, b.
- b. Các hệ số x, y.
- c. Các hệ số a, b, x, y.
- d. Các hệ số a, x, b.

Câu 111. Xây dựng lớp hình chữ nhật tạo bởi hai cạnh a, b. Để khai báo hàm tạo với tham số mặc định ta khai báo:

- a. `Hinhchunhat(float a=1, float b=1);`
- b. `Hinhchunhat(float a, float b);`
- c. `Hinhchunhat();`
- d. `Hinhchunhat(const Hinhchunhat &H);`

Câu 112. Cho đoạn chương trình như hình trên. Thuộc tính của lớp `Cong_dan` được khai báo là:

```
#include "conio.h"
#include "iostream.h"
#include "string.h"
#include "iomanip.h"
class Congdan
{
private:
    char ht[35], dc[50];
    char cmt[10];
public:
    Congdan()
    {
        strcpy(ht, "");
        strcpy(dc, "");
        strcpy(cmt, "");
    }
    Congdan(char ht[35], char dc[50], char cmt[10])
    {
        strcpy(this->ht, ht);
        strcpy(this->dc, dc);
    }
};
```

- a. `ht, dc, cmt`
- b. Không xác định
- c. `ht, cmt`
- d. `Congdan, ht, cmt, dc`

Câu 113. Một người nông dân thuộc quốc tịch Mỹ cả đời làm ra được khối tài sản tương ứng 10 triệu đô la. Để sinh tồn và làm việc ông tiêu tốn một khoản tiền tương ứng là 3 triệu đô; số tiền cũng như đất đai ông để lại cho con cháu là 5 triệu đô; 2 triệu đô còn lại là số mà ông đã và dự định làm từ thiện chi phí không mục đích với bạn bè. Như vậy ta có thể coi:

- a. Thông tin 3 triệu đô thuộc phạm vi `private`
- b. Thông tin 5 triệu đô thuộc phạm vi `protected`
- c. Thông tin 2 triệu đô thuộc phạm vi `public`
- d. Tất cả đều đúng

Câu 114. Để quản lý các phương tiện giao thông đường bộ người ta xây dựng hệ thống các lớp đối tượng gồm: lớp xe đạp, lớp xe máy, lớp ô tô con, lớp xe tải. Sau quá trình phân tích nhận thấy rằng các lớp đối tượng trên cùng có các đặc điểm như sau: Tải_trọng, loại_động_cơ, Biển_số và đều di chuyển từ điểm a đến điểm b. Người ta xây dựng lớp PT_Giao_Thông làm:



- a. Lớp dẫn xuất cho các lớp đối tượng trên.
- b. Lớp cơ sở cho các lớp đối tượng trên
- c. Lớp bao của các lớp đối tượng trên.
- d. Lớp thành viên của các lớp đối tượng trên.

Câu 115. Để quản lý các phương tiện giao thông đường bộ người ta xây dựng hệ thống các lớp đối tượng gồm: lớp xe đạp, lớp xe máy, lớp ô tô con, lớp xe tải. Sau quá trình phân tích nhận thấy rằng các lớp đối tượng trên cùng có các đặc điểm như sau: Tải_trọng, loại_động_cơ, Biển_số và đều di chuyển từ điểm a đến điểm b. Người ta xây dựng lớp PT_Giao_Thông làm lớp cơ sở cho các lớp đối tượng trên và lớp này gồm các thuộc tính là:



- a. Tải_trọng, Loại_động_cơ, Biển_số, Di_chuyển.
- b. Tải_trọng, Loại_động_cơ, Loại_phương_tiện, Biển_số.
- c. Tải_trọng, Loại_động_cơ, Di_chuyển.
- d. Tải_trọng, Loại_động_cơ, Loại_phương_tiện, Biển_số, Di_chuyển
- e. Tải_trọng, Loại_động_cơ, Biển_số.

Câu 116. Để quản lý các phương tiện giao thông đường bộ người ta xây dựng hệ thống các lớp đối tượng gồm: lớp xe đạp, lớp xe máy, lớp ô tô con, lớp xe tải. Sau quá trình phân tích nhận thấy rằng các lớp đối tượng trên cùng có các đặc điểm như sau: Tải_trọng, loại_động_cơ,

Biển_số và điều di chuyển từ điểm a đến điểm b. Người ta xây dựng lớp PT_Giao_Thông làm lớp cơ sở cho các lớp đối tượng trên và lớp này gồm các phương thức là:



- a. Di_chuyển.
- b. Tải_trọng, Loại_động_cơ, Loại_phương_tiện, Biển_số.
- c. Loại_phương_tiện, Di_chuyển.
- d. Tải_trọng, Loại_động_cơ, Loại_phương_tiện, Biển_số, Di_chuyển
- e. Tải_trọng, Loại_động_cơ, Biển_số.

Câu 117. Các trường hợp trùng tên trong kế thừa là:

- a. Trùng tên giữa phương thức lớp cha và phương thức lớp con; Trùng tên giữa phương thức giữa các lớp cha; Trùng tên giữa các lớp con.
- b. Trùng tên giữa phương thức lớp cha và phương thức lớp con; Trùng tên giữa phương thức giữa các lớp cha; Kế thừa lớp cha quá nhiều lần.
- c. Trùng tên giữa phương thức lớp cha và phương thức lớp con; Trùng tên giữa phương thức giữa các lớp cha;.
- d. Trùng tên giữa phương thức lớp cha và phương thức lớp con;; Trùng tên giữa các lớp con.

Câu 118. Cho lớp A, B, C trong đó lớp C kế thừa lớp A,B. Trong lớp A, B đều có phương thức Xuất, lớp C không có phương thức Xuất, khi khai báo đối tượng obj thuộc lớp C thì nếu gọi đến phương thức Xuất (obj.Xuat()) thì khi biên dịch thấy xuất hiện lỗi, trường hợp xảy ra lỗi này là do:

- a. Lớp A.
- b. Lớp B.
- c. Lỗi không thể xác định được.
- d. Trùng tên giữa phương thức các lớp cha

Câu 119. Người ta cần quản lý thông tin của công dân để triệu gọi nhập ngũ khi đến thời điểm, biết rằng điều kiện nhập ngũ là nam giới từ 18 tuổi trở lên không đang theo học tại bất kỳ trường đại học, cao đẳng nào. Phân tích đầu bài cho thấy, cần phải xây dựng:

- a. Lớp công dân và lớp Công dân nhập ngũ

b. Lớp công dân và lớp Công dân nam nhập ngũ

c. Lớp công dân

Câu 120. Khi đưa các lớp thực tế vào quản lý trong máy tính ta nhận thấy mỗi đối tượng thực tế có vô số thuộc tính nhưng với mỗi bài toán cụ thể ta chỉ xác định các lớp chỉ gồm một số thuộc tính nhất định. Cách thức đó gọi là:

a. Sự trừu tượng hóa dữ liệu.

b. Sự trừu tượng hóa chức năng.

c. Tính đa hình.

d. Tính kế thừa.

Câu 121. Vào mỗi kỳ thi người ta cần lập ra danh thí sinh dự thi dựa theo danh sách sinh viên đủ điều kiện dự thi của mỗi môn học, để thuận tiện xử lý người ta xây dựng lớp Thí sinh dựa trên lớp sinh viên với điều kiện là đủ điều kiện dự thi các môn học. Mối quan hệ giữa lớp Sinh viên và lớp thí sinh là:

a. Lớp Thí sinh là trường hợp đặc biệt hóa của lớp Sinh viên

b. Lớp Sinh viên là trường hợp đặc biệt hóa của lớp Thí sinh.

c. Lớp Thí sinh là trường hợp tổng quát của lớp Sinh viên.

d. Không có phương án đúng.

Câu 122. Khi xây dựng lớp trong C++, nếu trong lớp có thành phần thuộc tính là kiểu con trỏ hay tham chiếu thì bắt buộc phải:

a. Xây dựng hàm hủy

b. Xây dựng phương thức nhập

c. Xây dựng phương thức xuất.

d. Không bắt buộc phải xây dựng gì hết.

Câu 123. Xây dựng lớp điểm trong hệ toạ độ Oxyz các thuộc tính của lớp là:

a. Tung độ, hoành độ, cao độ

b. Tung độ, hoành độ, cao độ, dịch chuyển

c. Dịch chuyển

d. Không có phương án nào đúng.

Câu 124. Hãy cho biết thành phần thuộc tính của lớp đa thức xây dựng bằng mảng:

a. Bậc, mảng hệ số

b. Bậc, hệ số

c. Mảng bậc, mảng hệ số

d. Mảng bậc, hệ số.

Câu 125. Đoạn khai báo thuộc tính của lớp đa thức (với hệ số là một mảng số thực) là:

a. class Dathuc

```
{  
private:  
int bac;  
float *hs;  
};
```

b. class Dathuc

```
{  
private:  
int *bac;  
float *hs;  
};
```

c. class Dathuc

```
{  
private:  
int bac;  
float hs;  
};
```

d. class Dathuc

```
{  
private:  
int *bac;  
float hs;  
};
```

Câu 126. Vào mỗi kỳ thi người ta cần lập ra danh thí sinh dự thi dựa theo danh sách sinh viên đủ điều kiện dự thi của mỗi môn học, để thuận tiện xử lý người ta xây dựng lớp Thí sinh dựa trên lớp sinh viên với điều kiện là đủ điều kiện dự thi các môn học. Người ta phân tích thấy lớp thí sinh là trường hợp đặc biệt của lớp Sinh viên vì vậy khi xây dựng người ta xây dựng:

a. Lớp Thí sinh kế thừa lớp Sinh viên

b. Lớp Sinh viên kế thừa lớp Thí sinh.

c. Lớp Thí sinh là lớp bao của lớp Sinh viên.

d. Lớp Sinh viên là lớp bao của lớp Thí sinh

Câu 127. Cho lớp Dathuc, để thực hiện khai báo sau mà không lỗi:

```
void main()  
{  
Dathuc D1,D2(D1);  
}
```

Ta cần xây dựng:

- a. Hàm tạo không đối và hàm tạo sao chép;
- b. Hàm tạo không đối và hàm tạo có đối.
- c. Hàm tạo có đối số mặc định
- d. Hàm tạo có đối.

Câu 128. Xây dựng hàm tạo sao chép của lớp Diem trong C++ khai báo theo cú pháp sau:

- a. Diem (const Diem &D);
- b. Diem (Diem D);
- c. Diem (float x, float y);
- d. Diem (Diem D1, Diem D2);

Câu 129. Để tính khoảng cách hai điểm trong lớp điểm ta xây dựng phương thức tính khoảng cách hai điểm theo cú pháp sau: float KC2D(Diem D); Lời gọi đến phương thức để tính khoảng cách hai điểm là:

- a. cout<<D1.KC2D(D2);
- b. cout<<KC2D(D1,D2);
- c. KC2D(D1,D2).in()
- d. Không có phương án đúng

Câu 130. Trong kế thừa, lớp dẫn xuất có thể kế thừa các phương thức và thuộc tính của lớp cơ sở nhưng không:

- a. Kế thừa hàm tạo, toán tử gán
- b. Kế thừa hàm hủy, toán tử gán
- c. Kế thừa toán tử
- d. Tất cả các phương án trên.

Câu 131. Tính chất kế thừa chỉ ra rằng khi lớp A kế thừa lớp B thì:

- a. Lớp A sẽ có toàn bộ những thành phần thuộc protected và public của lớp B.
- b. Lớp B sẽ có toàn bộ những thành phần thuộc protected và public của lớp A.
- c. Lớp A sẽ có toàn bộ những thành phần thuộc private, protected và public của lớp B.
- d. Lớp A sẽ có toàn bộ những thành phần thuộc private và public của lớp B.

Câu 132. Trong lớp phân số khai báo: Phanso(int ts=0,int ms=1); Là :

- a. Hàm tạo có tham số mặc định của lớp Phanso
- b. Hàm tạo sao chép của lớp Phanso.
- c. Phương thức gán của lớp Phanso

d. Khai báo trên không thể tồn tại trong lớp Phanso

Câu 133. Hãy cho biết trong các ví dụ sau ví dụ nào thể hiện sự kế thừa :

a. Lớp lớp Người và Giáo Viên

b. Lớp Điểm và Hình tròn.

c. Lớp Xe ô tô và lớp Xe

d. Lớp Điểm và Lớp điểm màu

e. Tất cả các phương án đều đúng.

Câu 134. Cho lớp B, C kế thừa lớp A; Lớp D kế thừa lớp B và C. Khi đó đối tượng của lớp D không thể kế thừa được thành phần của A. Cần phải :

a. Khai báo A là lớp cơ sở ảo đối với B và C.

b. Khai báo lại A là lớp cơ sở ảo với D.

c. Khai báo A là lớp cơ sở ảo đối với B.

d. Hiển nhiên lớp D không thể kế thừa lớp A vì kế thừa không có tính bắc cầu.

Câu 135. Các phương thức cơ bản của một chiếc máy giặt là:

a. Cấp nước, Đảo trộn, Xả nước, Vắt khô

b. Đảo trộn, Xả nước, Vắt khô

c. Đảo trộn, Xoay, Vắt khô,

Câu 136. Khi xây dựng lớp đối tượng công việc cần làm là:

a. Xác định thành phần thuộc tính và phương thức của lớp đó.

b. Xác định thành phần private, public, protected là gì.

c. Xác định kiểu dữ liệu của thuộc tính, kiểu trả về của phương thức.

d. Tất cả các phương án đều đúng.

Câu 137. Khi khai báo lớp A là bạn của lớp B thì:

a. Chưa chắc lớp B đã là bạn của lớp A.

b. Chắc chắn lớp B là bạn của lớp A.

c. Lớp A sẽ kế thừa mọi thành phần của lớp B.

d. Tất cả các phương thức của lớp B đều có thể truy cập đến thành phần riêng của lớp A

Câu 138. Trường Đại học kinh doanh cũng in giấy báo nhập học vào mỗi kỳ tuyển sinh, Trường địa học Kinh tế quốc dân cũng in giấy báo nhập học, trường đại học Quốc gia cũng in Giấy báo nhập học, tuy nhiên 3 giấy báo nhập học này lại có thể có cấu trúc, nội dung, hình thức khác nhau. Ví dụ này chứng tỏ tính:

- a. Đa hình
- b. Áo của phương thức.
- c. Trừu tượng
- d. Đóng gói.

Câu 139. Một lớp có thể:

- a. Là lớp cơ sở của vô số lớp.
- b. Là lớp bạn của vô số lớp.
- c. Là lớp dẫn xuất của vô số lớp.
- d. Tất cả các phương án đều đúng.

Câu 140. Chồng hàm là:

- a. Khả năng các hàm có thể trùng tên nhau nhưng khác nhau về: kiểu dữ liệu trả về hoặc kiểu dữ liệu của tham số hoặc số lượng tham số truyền vào.
- b. Khả năng các hàm có thể trùng tên nhau, trùn cả về: kiểu dữ liệu trả về, kiểu dữ liệu của tham số, số lượng tham số truyền vào.
- c. Khả năng các hàm có thể trùng tên nhau nhưng khác nhau về: kiểu dữ liệu trả về hoặc kiểu dữ liệu của tham số hoặc số lượng tham số truyền vào. Chỉ có thể chồng hàm thành viên của lớp
- d. Tất cả các phương án đều đúng.

Câu 141. Xây dựng lớp `Hinh_hoc` là lớp cơ sở cho các lớp `Hinh_vuong`, `Hinh_tron`, `Hinh_cn`, `Hinh_tamgiac`. Nhận thấy các lớp này đều có phương thức tính `D_Tich`(diện tích). Tuy nhiên cách tính diện tích của các phương thức `D_Tich` ứng với các đối tượng của các lớp dẫn xuất là khác nhau. Điều này thể hiện tính:

- a. Đa hình
- b. Áo của phương thức.
- c. Trừu tượng
- d. Đóng gói.

Câu 142. Các lớp có thể kế thừa từ lớp động vật là:

- a. Lớp Con chó (1)
- b. Lớp Con mèo (2)
- c. Lớp cây cối (3)
- d. Cả (1) và (2)

Câu 143. Các lớp có thể kế thừa từ lớp Điện thoại

- a. Lớp Iphone
- b. Lớp điện thoại LG
- c. Lớp điện thoại di động
- d. Lớp điện thoại cố định
- e. Tất cả phương án

Câu 144. Các phương thức cơ bản của lớp xe hơi là:

- a. Khởi động, Chuyển bánh, Rẽ phải, Rẽ trái, Di chuyển, Quay đầu, Lùi sau
- b. Khởi động, Chuyển bánh, Rẽ phải, Rẽ trái, Di chuyển, Lùi sau.
- c. Khởi động, Di chuyển, Rẽ phải, Rẽ trái, Lùi sau.
- d. Không có phương thức nào là của lớp xe hơi.

Câu 145. Bắt buộc phải xây dựng hàm tạo lớp khi:

- a. Khi cần khởi gán giá trị ban đầu cho thành phần thuộc tính của lớp
- b. Khi cần khởi gán giá trị ban đầu cho thành phần phương thức của lớp
- c. Khi cần truyền tham số cho hàm thành viên của lớp.
- d. Không bắt buộc phải xây dựng hàm tạo

Câu 146. Cho đoạn chương trình sau:

```
class Phanso
{
private:
int ts,ms;
public:
Phanso(int ts1, int ms1)
{ts=ts1;ms=ms1;}
};
void main()
{
Phan so P1,P2(3,4);
}
```

Câu lệnh khai báo Phanso P1 bị báo lỗi vì:

- a. Gọi tới hàm tạo không đối mà chương trình lại không xây dựng.
- b. Chương trình trên không thể lỗi được vì chương trình C++ sẽ tự phát sinh hàm tạo không đối.
- c. Không xác định được gọi tới hàm tạo nào
- d. Tất cả đều sai.

Câu 147. Mối quan hệ giữa lớp Người và lớp Sinh viên là:

- a. Lớp Người là lớp cơ sở cho lớp Sinh Viên.
- b. Lớp Người là lớp dẫn xuất của lớp Sinh Viên
- c. Lớp Người là lớp bao của lớp Sinh Viên
- d. Lớp Người là lớp đối tượng thành phần của lớp Sinh Viên.

Câu 148. Mối quan hệ giữa lớp Môn Học và lớp Sinh viên là:

- a. Lớp Môn Học là lớp cơ sở cho lớp Sinh Viên.
- b. Lớp Môn Học là lớp dẫn xuất của lớp Sinh Viên
- c. Lớp Môn Học là lớp bao của lớp Sinh Viên
- d. Lớp Môn Học là lớp đối tượng thành phần của lớp Sinh Viên.

Câu 149. Khi xây dựng lớp đối tượng, các thuộc tính dữ liệu thông thường được khai báo trong phạm vi:

- a. Private
- b. Public
- c. Private và Protected
- d. Protected

Câu 150. Cho đoạn chương trình:

```
class Diem
{
float x,y;
public:
Diem(){x=0;y=0}; //ham tao khong doi
Diem(float x,float y);//ham tao co doi
};
class HT:public Diem
{
private:
float r;
public:
HT(){r=1;};
HT(float x1,float y1, float r1){r=r1;};
};
```

Đoạn chương trình trên khi khai báo 1 hình tròn (HT H1;) thì hình tròn H1 này có:

- a. Không xác định được tâm vì theo tính chất kế thừa không kế thừa hàm tạo, ở lớp dẫn xuất phải gọi hàm tạo của lớp cơ sở.
- b. Tâm là (0,0) bán kính là 1
- c. Tâm là(1,1) bán kính là 1

d. Tất cả các phương án đều đúng.

Câu 151. Cho đoạn chương trình:

```
class Diem
{
float x,y;
public:
Diem(){x=0;y=0}; //ham tao khong doi
Diem(float x,float y);//ham tao co doi
};
class HT:public Diem
{
private:
float r;
public:
HT(){r=1;};
HT(float x1,float y1, float r1){r=r1;};
};
```

Đoạn chương trình trên khi khai báo 1 hình tròn (HT H1;) thì hình tròn H1 này chưa xác định được tâm vì theo tính chất kế thừa không kế thừa hàm tạo ta phải tự triệu gọi hàm tạo của lớp cơ sở ở lớp dẫn xuất. Để xác định tâm cho hình tròn H1 là 0,0 ta cần sửa là:

a. HT():Diem(){r=1;};

b. HT(){x=0,y=0;r=1;};

c. HT(){Diem();r=1;};

d. Tất cả các phương án đều đúng.

Câu 152. Để đưa đối tượng trong thực tế vào máy tính ta cần chú trọng đến tính:

a. Trừu tượng dữ liệu và trừu tượng chức năng

b. Đa hình của nó.

c. Kế thừa.

d. Bao gói.

Câu 153. Khi khai báo, xây dựng lớp các thuộc tính của lớp thường được khai báo trong phạm vi Private, điều này thể hiện đặc tính:

a. Bao gói

b. Trừu tượng

c. Đa hình

d. Chồng hàm

Câu 154. Khi khai báo, xây dựng lớp các phương thức thường được khai báo trong phạm vi public để:

- a. Tương tác với các lớp hay môi trường bên ngoài.
- b. Thể hiện tính bao gói dữ liệu
- c. Thể hiện rõ tính chất đa hình.
- d. Tương tác với các thuộc tính bên trong lớp.

Câu 155. Cho lớp Time với ngôn ngữ C++ như sau:

```
class time {
int hour,min, sec;
public:
void addHour(int h);
}
```

Khi xây dựng phương thức addHour bên ngoài lớp Time ta phải viết là:

- a. void addHour(int h){ //Nội dung}
- b. void addHour(int h);{//Nội dung}
- c. void time:addHour(int h){//Nội dung}
- d. void time::addHour(int h){//Nội dung}

Câu 156. Cho lớp Điểm như sau:

Trong lớp Điểm trên thì KC2D có tên gọi là:

```
class diem
{
    private:
        int x,y;//toa do x va toa do
        y
    public:
        void nhapd();//phuong
        thuc nhap
        friend float KC2D(diem D1,
        diem D2)
};
```

- a. Hàm bạn của lớp Điểm
- b. Phương thức của lớp
- c. Hàm thông thường
- d. Toán tử bạn của lớp Điểm

Câu 157. Cho lớp Điểm như trên, Trong lớp Điểm trên thì hàm KC2D có tên gọi là:

```

class diem
{
    private:
        int x,y;//toa do x va toa do
        y
    public:
        void nhapd();//phuong
        thuc nhap
        friend float KC2D(diem D1,
            diem D2)

        ,};

```

- a. Hàm bạn của lớp Điểm
- b. Phương thức của lớp
- c. Hàm thông thường
- d. Toán tử bạn của lớp Điểm

Câu 158. Cho lớp Điểm như sau:

Trong lớp Điểm trên thì để gọi tới hàm KC2D ta sử dụng câu lệnh:

```

class diem
{
    private:
        int x,y;//toa do x va toa do
        y
    public:
        void nhapd();//phuong
        thuc nhap
        friend float KC2D(diem D1,
            diem D2)

        ,};

```

- a. D1.KC2D(D2);
- b. D2.KC2D(D1);
- c. KC2D(D1,D2);
- d. D1::KC2D(D2);

Câu 159. Cho khai báo sau:

```

class A;          L
class B ;
class C;
// Định nghĩa các lớp
class A
{
...
friend class B ; // Lớp B là bạn của A
friend class C ; // Lớp C là bạn của A
...
};

```

Từ khai báo trên, câu khẳng định sai là:

- a. Lớp A là bạn của lớp B
- b. Lớp C là bạn của lớp A
- c. Các phương thức của lớp B có quyền truy cập đến tất cả các thành viên riêng của lớp A
- d. Các phương thức của lớp A có quyền truy cập đến tất cả các thành viên riêng của lớp B

Câu 160. Khi khai báo, xây dựng lớp các thành phần được đặt trong phạm vi protected nhằm mục đích:

- a. Cho phép kế thừa nhưng không cho phép tương tác trực tiếp từ bên ngoài lớp.
- b. Cho phép kế thừa và cho phép tương tác trực tiếp từ bên ngoài lớp.
- c. Chỉ cho phép kế thừa ngay bên trong lớp đó cũng không truy cập được.

Câu 161. Xây dựng lớp Time trong java như hình trên, phương thức là:

```
import java.text.DecimalFormat;
public class Time
{
    private int hour;    // 0 - 23
    private int minute;  // 0 - 59
    private int second;  // 0 - 59
    // Câu tu
    public Time() { setTime( 0, 0, 0 ); }
    // Hàm thiết lập dữ liệu
    public void setTime( int h, int m, int s )
    {
        hour = ( ( h >= 0 && h < 24 ) ? h : 0 );
        minute = ( ( m >= 0 && m < 60 ) ? m : 0 );
        second = ( ( s >= 0 && s < 60 ) ? s : 0 );
    }
}
```

- a. Phương thức khởi tạo của lớp Time
- b. Phương thức nhập dữ liệu cho lớp Time
- c. Phương thức kiểm tra thời gian của lớp Time

Câu 162. Xây dựng lớp Time trong java như hình trên, phương thức khởi tạo public Time() làm nhiệm vụ:


```

import java.text.DecimalFormat;
public class Time
{
    private int hour;    // 0 - 23
    private int minute;  // 0 - 59
    private int second;  // 0 - 59
    // Cau tu
    public Time() { setTime( 0, 0, 0 ); }
    // Ham thiet lap du lieu
    public void setTime( int h, int m, int s )
    {
        hour = ( ( h >= 0 && h < 24 ) ? h : 0 );
        minute = ( ( m >= 0 && m < 60 ) ? m : 0 );
        second = ( ( s >= 0 && s < 60 ) ? s : 0 );
    }
}

```

- Gọi tới hàm setTime để thiết lập giá trị khởi tạo cho giờ, phút, giây
- Gọi tới hàm setTime để hủy bỏ toàn bộ dữ liệu thuộc tính của lớp hour
- Khởi tạo giá trị ban đầu cho dữ liệu của lớp.

Câu 163. Xây dựng lớp Time trong java như hình trên, phương thức khởi tạo public Time() làm nhiệm vụ:

```

import java.text.DecimalFormat;
public class Time
{
    private int hour;    // 0 - 23
    private int minute;  // 0 - 59
    private int second;  // 0 - 59
    // Cau tu
    public Time() { setTime( 0, 0, 0 ); }
    // Ham thiet lap du lieu
    public void setTime( int h, int m, int s )
    {
        hour = ( ( h >= 0 && h < 24 ) ? h : 0 );
        minute = ( ( m >= 0 && m < 60 ) ? m : 0 );
        second = ( ( s >= 0 && s < 60 ) ? s : 0 );
    }
}

```

- a. Gọi tới hàm setTime để thiết lập giá trị khởi tạo cho giờ, phút, giây
- b. Gọi tới hàm setTime để hủy bỏ toàn bộ dữ liệu thuộc tính của lớp hour
- c. Khởi tạo giá trị ban đầu cho dữ liệu của lớp.

Câu 164. Cho đoạn chương trình trên, câu nhận định đúng cho đoạn chương trình trên là:

```

class Sample
{
    private int i;
    public Single j;
    private void DisplayData()
    {
        Console.WriteLine(i + " " + j);
    }
    public void ShowData()
    {
        Console.WriteLine(i + " " + j);
    }
}

```

- a. DisplayData() không thể được khai báo với từ khóa private
- b. DisplayData() không thể truy cập vào j
- c. ShowData() không thể truy cập vào i
- d. Đoạn chương trình không lỗi

Câu 165. Đoạn chương trình trên làm nhiệm vụ:

```

template <class T> T giatrimax(T a, T b)
{
    if(a>b)
        return a;
    else
        return b;
}

```

- a. Khai báo và xây dựng template tìm giá trị lớn nhất của hai phần tử có giá trị dữ liệu bất kỳ giống nhau.
- b. Khai báo và xây dựng template tìm giá trị nhỏ nhất của hai phần tử có giá trị dữ liệu bất kỳ khác nhau.
- c. Khai báo và xây dựng template tìm giá trị lớn nhất của hai phần tử có giá trị dữ liệu bất kỳ khác nhau.

Câu 166. Cho bảng Lớp trong CSDL sql server, thực hiện kết nối đến CSDL và lấy dữ liệu về bảng lớp để hiển thị lên Form. Ta cần xây dựng lớp Lớp học trong C# có cấu trúc như hình trên. Thuộc tính của lớp Lớp học:

```
public class Lophoc
{
    #region Fields
    private string connString;
    public string ConnectionStr...
    #endregion
    #region Properties
    private string maLop;
    public string MaLop...
    private string tenLop;
    public string TenLop...
    private int namVao;
    public int NamVao...
    private string maKhoa;
    public string MaKhoa...
    #endregion
    public Lophoc(string str)...
    public DataTable getClasses()...
    public DataTable getClassesInFaculty()...
    public bool deleteClass()...
    public bool updateClass()...
```

- a. malop, tenlop,namvao, makhoa
- b. malop,tenlop,connString, namvao,makhoa
- c. Malop, Tenlop, Namvao, Makhoa
- d. Lophoc, deleteClass, updateClass

Câu 167. Cho lớp Lophoc như hình trên, Lớp Lophoc dùng để kết nối đến bảng Lop trong CSDL SQL Server và lấy dữ liệu và thực thi một số thao tác như update, delete, insert... Ý nghĩa của dòng lệnh “private string malop;” là:

```
public class Lophoc
{
    #region Fields
    private string connString;
    public string ConnectionStr[...]
    #endregion
    #region Properties
    private string maLop;
    public string MaLop[...]
    private string tenLop;
    public string TenLop[...]
    private int namVao;
    public int NamVao[...]
    private string maKhoa;
    public string MaKhoa[...]
    #endregion
    public Lophoc(string str)[...]
    public DataTable getClasses()[...]
    public DataTable getClassesInFaculty()[...]
    public bool deleteClass()[...]
    public bool updateClass()[...]
```

- a. Khai báo thuộc tính maLop kiểu chuỗi
- b. Khai báo phương thức trả về maLop kiểu chuỗi
- c. Khai báo hàm tạo của lớp Lophoc
- d. Không có phương án chính xác

Câu 168. Cho lớp Lophoc như hình trên , ý nghĩa của phương thức “ public string Malop” là:

```

public class Lophoc
{
    #region Fields
    private string connString;
    public string ConnectionStr[...]
    #endregion
    #region Properties
    private string maLop;
    public string MaLop[...]
    private string tenLop;
    public string TenLop[...]
    private int namVao;
    public int NamVao[...]
    private string maKhoa;
    public string MaKhoa[...]
    #endregion
    public Lophoc(string str)[...]
    public DataTable getClasses()[...]
    public DataTable getClassesInFaculty()[...]
    public bool deleteClass()[...]
    public bool updateClass()[...]
}

```

- a. Dùng để thiết lập và lấy giá trị của thuộc tính maLop(2)
- b. Khai báo phương thức Malop trả về kiểu chuỗi
- c. Dùng để truy cập vào thuộc tính maLop của lớp (1)
- d. Cả (1) và (2)

Câu 169. Khi khai báo các thành phần thuộc tính của lớp trong C#, đầu mỗi câu lệnh khai báo ta thường khai báo:

- a. Phạm vi của thuộc tính là private hay public hay protected

- b. Khai báo kiểu dữ liệu của thuộc tính
- c. Từ khóa Properties đi đầu
- d. Từ khóa Region đi đầu

Câu 170. Cấu trúc thông thường của lớp trong C# là:

- a. Khai báo các trường dữ liệu cần dùng, Khai báo các thuộc tính và các phương thức thiết lập, lấy giá trị của thuộc tính, Khai báo và xây dựng các phương thức của lớp.
- b. Khai báo các trường dữ liệu cần dùng, Khai báo các thuộc tính, Khai báo và xây dựng các phương thức của lớp.
- c. Khai báo các thuộc tính và các phương thức thiết lập, lấy giá trị của thuộc tính, Khai báo và xây dựng các phương thức của lớp.
- d. Khai báo các trường dữ liệu cần dùng, Khai báo các thuộc tính Khai báo các phương thức của lớp.

Câu 171. Cho lớp Lophoc được khai báo như trên, khi thực hiện khai báo đối tượng của lớp Lophoc: Lophoc obj1=new Lophoc('Chuỗi kết nối'). Để thiết lập thuộc tính maLop của đối tượng obj1 ta sử dụng câu lệnh:

```

public class Lophoc
{
    #region Fields
    private string connString;
    public string ConnectionStr[...]
    #endregion
    #region Properties
    private string maLop;
    public string MaLop[...]
    private string tenLop;
    public string TenLop[...]
    private int namVao;
    public int NamVao[...]
    private string maKhoa;
    public string MaKhoa[...]
    #endregion
    public Lophoc(string str)[...]
    public DataTable getClasses()[...]
    public DataTable getClassesInFaculty()[...]
    public bool deleteClass()[...]
    public bool updateClass()[...]
}

```

- Không thể thiết lập được vì maLop là thành viên kiểu private
- Thực hiện truy cập thông qua phương thức Malop đã xây dựng như sau: obj1.Malop=...
- Chưa khai báo phương thức thiết lập maLop
- Thực hiện truy cập như sau: obj1.maLop=....

Câu 172. Thông thường khi xây dựng lớp trong C#, với mỗi khai báo thuộc tính của lớp ta đều xây dựng thêm hai phương thức set và get dùng để:

- Thiết lập và lấy giá trị của thuộc tính đó.

- b. Tăng tính bảo mật dữ liệu của thuộc tính.
- c. Để nhập và xuất giá trị thuộc tính đó.
- d. Đây là cấu trúc yêu cầu của C# khi khai báo thuộc tính của lớp

Câu 173. Cho lớp Lophoc được khai báo như trên, nếu thay câu lệnh private string maLop thành public string maLop thì:

```
public class Lophoc
{
    #region Fields
    private string connString;
    public string ConnectionStr...
    #endregion
    #region Properties
    private string maLop;
    public string MaLop...
    private string tenLop;
    public string TenLop...
    private int namVao;
    public int NamVao...
    private string maKhoa;
    public string MaKhoa...
    #endregion
    public Lophoc(string str)...
    public DataTable getClasses()...
    public DataTable getClassesInFaculty()...
    public bool deleteClass()...
    public bool updateClass()...
```

- a. Điều này là không được phép. Vì quy định trong C# các thuộc tính phải khai báo bắt đầu bằng từ khóa private.
- b. Thuộc tính sẽ có phạm vi Public, được truy cập thoải mái từ bên ngoài.(1)

c. Việc xây dựng phương thức public string Malop là không cần thiết nữa.(2)

d. Cả (1) và (2)

Câu 174. Cho lớp Lophoc được khai báo như trên, với khai báo public Lophoc (string str); là khai báo:

```
public class Lophoc
{
    #region Fields
    private string connString;
    public string ConnectionStr...
    #endregion
    #region Properties
    private string maLop;
    public string MaLop...
    private string tenLop;
    public string TenLop...
    private int namVao;
    public int NamVao...
    private string maKhoa;
    public string MaKhoa...
    #endregion
    public Lophoc(string str)...
    public DataTable getClasses()...
    public DataTable getClassesInFaculty()...
    public bool deleteClass()...
    public bool updateClass()...
```

a. Hàm tạo của lớp.(1)

b. Một phương thức xác định chuỗi kết nối cho lớp.(2)

c. Cả (1) và (2)

d. Thuộc tính của lớp.

Câu 175. Cho lớp Lophoc được khai báo như trên, ý nghĩa của việc xây dựng phương thức `public DataTable getClasses()` là:

```
public class Lophoc
{
    #region Fields
    private string connString;
    public string ConnectionStr...
    #endregion
    #region Properties
    private string maLop;
    public string MaLop...
    private string tenLop;
    public string TenLop...
    private int namVao;
    public int NamVao...
    private string maKhoa;
    public string MaKhoa...
    #endregion
    public Lophoc(string str)...
    public DataTable getClasses()...
    public DataTable getClassesInFaculty()...
    public bool deleteClass()...
    public bool updateClass()...
```

- a. Thực hiện truy vấn CSDL, trả về bảng Lop
- b. Thực hiện thiết lập giá trị cho các thuộc tính của lớp
- c. Không có phương án đúng

Câu 176. Cần In danh sách thí sinh dự thi theo phòng gồm các thông tin: SBD, Họ và tên, Ngày sinh, Giới tính, Phòng thi, Giờ thi. Người ta xây dựng hai lớp đối tượng là lớp đối tượng Thí

sinh, lớp đối tượng Danh sách thí sinh theo phòng. Thuộc tính của lớp Danh sách thí sinh có thể là:

- a. Số thí sinh, mảng các thí sinh
- b. SBD, Họ và tên, Ngày sinh, Giới tính, Phòng thi, Giờ thi
- c. Mảng SBD, mảng Họ và tên, mảng Ngày sinh, Mảng Giới tính, Mảng Phòng thi, Mảng Giờ thi.
- d. Tất cả các phương án gộp lại

Câu 177. Cần In danh sách thí sinh dự thi theo phòng gồm các thông tin: SBD, Họ và tên, Ngày sinh, Giới tính, Phòng thi, Giờ thi. Người ta xây dựng hai lớp đối tượng là lớp đối tượng Thí sinh, lớp đối tượng Danh sách thí sinh theo phòng. Thuộc tính của lớp thí sinh là:

- a. Số thí sinh, mảng các thí sinh
- b. SBD, Họ và tên, Ngày sinh, Giới tính, Phòng thi, Giờ thi
- c. Mảng SBD, mảng Họ và tên, mảng Ngày sinh, Mảng Giới tính, Mảng Phòng thi, Mảng Giờ thi.
- d. Tất cả các phương án gộp lại

Câu 178. Cần In danh sách thí sinh dự thi gồm các thông tin: SBD, Họ và tên, Ngày sinh, Giới tính, Phòng thi, Giờ thi. Người ta xây dựng hai lớp đối tượng là lớp đối tượng Thí sinh, lớp đối tượng Danh sách thí sinh theo phòng. Phương thức cần thiết theo yêu cầu của bài toán cần phải xây dựng cho lớp thí sinh là:

- a. Phương thức nhập và hiển thị thông tin từng thí sinh
- b. Phương thức hiển thị thông tin từng thí sinh
- c. Phương thức hiển thị, phương thức khởi tạo, phương thức nhập
- d. Không có phương án đúng

Câu 179. Cần In danh sách thí sinh dự thi gồm các thông tin: SBD, Họ và tên, Ngày sinh, Giới tính, Phòng thi, Giờ thi. Người ta xây dựng hai lớp đối tượng là lớp đối tượng Thí sinh, lớp đối tượng Danh sách thí sinh theo phòng. Phương thức cần thiết theo yêu cầu của bài toán cần phải xây dựng cho lớp danh sách thí sinh là:

- a. Phương thức nhập và hiển thị danh sách thí sinh
- b. Phương thức hiển thị danh sách thí sinh
- c. Phương thức hiển thị, phương thức khởi tạo, phương thức nhập
- d. Không có phương án đúng

Câu 180. Cho hình ảnh trên, hình ảnh gợi ý cho ta về:

```
void myMethod( int p1 );  
void myMethod( int p1, int p2 );  
void myMethod( int p1, string p2 );
```

- a. Hiện tượng nạp chồng phương thức, hàm
- b. Điều không tồn tại trong lập trình hướng đối tượng
- c. Điều không tồn tại trong lập trình cấu trúc

Câu 181. Cho đoạn chương trình như hình trên, lớp Convat được gọi là:

```
class Convat
{
protected:
char *ten ;
public:
Convat()
{
ten=NULL;
}
Convat (char *ten1)
{
ten=strdup(ten1);
}
virtual void xungten() { ... }
};
class Concho { ... };
class Conmeo { ... };
```

- a. Lớp cơ sở
- b. Lớp cơ sở trừu tượng
- c. Lớp cơ sở ảo
- d. Lớp dẫn xuất

Câu 182. Cho đoạn chương trình như hình trên, dòng khai báo virtual void xungten() là:

```

class Convat
{
protected:
char *ten ;
public:
Convat()
{
ten=NULL;
}
Convat (char *ten1)
{
ten=strdup(ten1);
}
virtual void xungten() { ... }
};
class Concho { ... };
class Conmeo { ... };

```

- Khai báo một phương thức ảo
- Khai báo một phương thức tĩnh
- Khai báo một phương thức riêng không cho phép kế thừa
- Khai báo một phương thức bắt buộc phải có trong các lớp kế thừa

Câu 183. Để khai báo thành viên tĩnh trong C++ và Java đều dùng từ khóa static. Sự nhận định các phương thức tĩnh trong C++ và trong Java là:

- Các phương thức trong C++ không khai báo từ khóa static vẫn mặc định là thành viên tĩnh (1).
- Các phương thức trong Java không khai báo từ khóa static mặc định là phương thức ảo (2)
- Cả (1) và (2)
- Sự nhận định thành viên tĩnh trong Java và C++ là như nhau.

Câu 184. Cho đoạn chương trình khai báo như trên, việc sử dụng từ khóa static đứng trước khai báo tongsohd, tongtienban có ý nghĩa:

```
class HDBH
{
private:
char *tenhang;
float tienban;
static float tongsohd; //s
static float tongtienban;
```

- a. Khai báo các thuộc tính tĩnh, dùng chung bộ nhớ với các đối tượng khác nhau.
- b. Khai báo các thuộc tính tĩnh, dùng riêng bộ nhớ với các đối tượng khác nhau
- c. Khai báo các thuộc tính ảo, dùng chung bộ nhớ với các đối tượng khác nhau.
- d. Khai báo các thuộc tính ảo, dùng riêng bộ nhớ với các đối tượng tham gia kế thừa.

Câu 185. Một bạn xây dựng lại lớp số nguyên như trên, tại dòng số 3 bạn khai báo:

```

class son
{
1  int gt;
2  public:
3  son();
4  son(const int &n) { ... }
5  son(const son &n);
6  friend ostream &operator <<(ost
7  friend istream & operator>>(ist
8  son operator+(son b) { ... }
9  son operator- ();
10 son operator- (son b);
11 son operator* (son b);
12 son operator/ (son b);
13 son operator! ();
14 son operator++ ();
15 son operator-- ();
16 son operator^ (son mu);
};

```

- Khai báo hàm tạo không đối của lớp số nguyên
- Khai báo thuộc tính của lớp số nguyên
- Khai báo phương thức nhập giá trị của lớp số nguyên
- Khai báo phương thức hiển thị giá trị của lớp số nguyên.

Câu 186. Một bạn xây dựng lại lớp số nguyên như trên, tại dòng số 5 bạn khai báo:


```

class son
{
1  int gt;
2  public:
3  son();
4  son(const int &n) { ... }
5  son(const son &n);
6  friend ostream &operator <<(ost
7  friend istream & operator>>(ist
8  son operator+(son b) { ... }
9  son operator- ();
10 son operator- (son b);
11 son operator* (son b);
12 son operator/ (son b);
13 son operator! ();
14 son operator++ ();
15 son operator-- ();
16 son operator^ (son mu);
};

```

- Khai báo hàm tạo sao chép của lớp số nguyên
- Khai báo thuộc tính của lớp số nguyên
- Khai báo phương thức nhập giá trị của lớp số nguyên
- Khai báo hàm tạo không đối.

Câu 187. Một bạn xây dựng lại lớp số nguyên như trên, Bạn có thực hiện khai báo toán tử tăng trước tại dòng số:

```

class son
{
1  int gt;
2  public:
3  son();
4  son(const int &n) { ... }
5  son(const son &n);
6  friend ostream &operator <<(ost
7  friend istream & operator>>(ist
8  son operator+(son b) { ... }
9  son operator-();
10 son operator-(son b);
11 son operator*(son b);
12 son operator/(son b);
13 son operator!();
14 son operator++();
15 son operator--();
16 son operator^(son mu);
};

```

- a. 14
- b. 15
- c. 16
- d. 13

Câu 188. Một bạn xây dựng lại lớp số nguyên như trên, bạn có thực hiện khai báo toán tử giảm sau tại dòng số:

```

class son
{
1  int gt;
2  public:
3  son();
4  son(const int &n) { ... }
5  son(const son &n);
6  friend ostream &operator <<(ost
7  friend istream & operator>>(ist
8  son operator+(son b) { ... }
9  son operator- ();
10 son operator- (son b);
11 son operator* (son b);
12 son operator/ (son b);
13 son operator! ();
14 son operator++ ();
15 son operator-- ();
16 son operator^ (son mu);
};

```

- a. 14
- b. 15
- c. 16
- d. Bạn đó không khai báo toán tử giảm sau

Câu 189. Một bạn xây dựng lại lớp số nguyên như trên, bạn đó có thực hiện xây dựng nhập xuất cho số nguyên. Nếu có son A; ta có thể thực gọi nhập số nguyên A này bằng cách:

```

class son
{
1  int gt;
2  public:
3  son();
4  son(const int &n) { ... }
5  son(const son &n);
6  friend ostream &operator <<(ost
7  friend istream & operator>>(ist
8  son operator+(son b) { ... }
9  son operator-();
10 son operator-(son b);
11 son operator*(son b);
12 son operator/(son b);
13 son operator!();
14 son operator++();
15 son operator--();
16 son operator^(son mu);
};

```

- a. cin>>A;
- b. A.Nhap();
- c. A.cin();
- d. A>>

Câu 190. Một bạn xây dựng lại lớp đường thẳng $AX+BY+C=0$ như trên, và bạn đó có xây dựng hàm tạo cho lớp tại dòng số 4. Hãy xác định loại hàm tạo mà bạn đó đã xây dựng:

```
class dthang
```

```
1 {  
2 float A,B,C;  
3 public:  
4     dthang(float A=0,float B=0,f  
5     friend ostream&operator<<(ost  
6     friend istream &operator>>(is  
7     float hsogoc();  
8     float Kc2dth(dthang d);  
9     float kc1d(float x,float y);  
    };
```

- a. Hàm tạo không đối
- b. Hàm tạo sao chép
- c. Hàm tạo có tham số mặc định
- d. Không có phương án đúng

Câu 191. Một bạn xây dựng lại lớp đường thẳng $AX+BY+C=0$ như trên, và bạn đó có xây dựng hàm tạo cho lớp tại dòng số 4. Ta nhận thấy các tham số truyền vào có tên trùng với tên thuộc tính của lớp (A,B,C). Để phân biệt chúng ta sử dụng:

```

class dthang
1 {
2 float A,B,C;
3 public:
4     dthang(float A=0,float B=0,f
5     friend ostream&operator<<(ost
6     friend istream &operator>>(is
7     float hsogoc();
8     float Kc2dth(dthang d);
9     float kc1d(float x,float y);
    };

```

- Con trỏ this
- Không thể phân biệt được phải khai báo lại tên tham số truyền vào
- Cứ viết như bình thường (A=A) chương trình tự xác định được
- Không có phương án đúng

Câu 192. Một bạn xây dựng lớp Circle(hình tròn) kế thừa lớp point (điểm) như trên. Một bạn nhận xét chương trình sẽ bị lỗi tại dòng số 6 vì bạn chưa khai báo lớp point. Theo bạn nhận xét này:

#include<iostream.h>	1
#include<conio.h>	2
#include<stdio.h>	3
#include<diem.h>	4
#include<math.h>	5
class cicltre:public point	6
{	7
private:	8
float r;	9
public:	10
cicltre(float x=0, float y=0,	11
float getR());	12
friend ostream &operator<< (13
friend istream &operator>> (14
void setR(float r);	15
float cV();	16
float dT();	17
};	18

- Nhận xét rất chính xác
- Chương trình không lỗi vì lớp point đã được xây dựng tại thư viện diem.h ở dòng số 4
- Chương trình sẽ không lỗi dù lớp point chưa được xây dựng.
- Không có phương án đúng

Câu 193. Một bạn sinh viên xây dựng lớp Cicltre(hình tròn) kế thừa lớp point (điểm) như trên. Để xác định đường tròn bạn cần một điểm (tâm) và bán kính, tuy nhiên khi khai báo hình tròn bạn sinh viên này chỉ khai báo mỗi bán kính r vì:

#include<iostream.h>	1
#include<conio.h>	2
#include<stdio.h>	3
#include<diem.h>	4
#include<math.h>	5
class cicltre:public point	6
{	7
private:	8
float r;	9
public:	10
cicltre(float x=0, float y=0,	11
float getR());	12
friend ostream &operator<< (13
friend istream &operator>> (14
void setR(float r);	15
float cV();	16
float dT();	17
};	18

- Vì điểm (tâm) được kế thừa sẵn từ lớp point(điểm)
- Bạn này đã xác định sai thuộc tính của lớp
- Vì lớp đường tròn (Cicltre) mặc định là bạn của lớp điểm (point).

Câu 194. Một bạn xây dựng lớp Cicltre(hình tròn) kế thừa lớp point (điểm) như trên. Để nhập dữ liệu cho lớp Hình tròn bạn đó đã sử dụng:

#include<iostream.h>	1
#include<conio.h>	2
#include<stdio.h>	3
#include<diem.h>	4
#include<math.h>	5
class ciclre:public point	6
{	7
private:	8
float r;	9
public:	10
ciclre(float x=0,float y=0,	11
float getR());	12
friend ostream &operator<< (13
friend istream &operator>> (14
void setR(float r);	15
float cV();	16
float dT();	17
};	18

- a. Phương thức Nhap()
- b. Toán tử nhập cin>>
- c. Hàm bạn toán tử nhập cin>>
- d. Bạn này không khai báo nhập dữ liệu cho lớp Hình tròn

Câu 195. Một bạn sinh viên xây dựng cấu trúc Stack dưới dạng khai báo lớp như trên, Bạn sinh viên đã khai báo xây dựng stack lưu trữ dưới dạng:

```
class stack
{
    private:
        int *data;
        int size;
        int top;
    public:
        stack(int size);
        void push(int x);
        void pop(int &x);
        int isFull();
        int isEmpty();
        ~stack();
};
```

- a. Mảng
- b. Danh sách liên kết
- c. Không xác định được

Câu 196. Nếu A và B là hai lớp thì với khai báo lớp C như trên thì lớp C được gọi là:

```

class C
{
    private:
        int m, n;
        A u, v;
        B p, q, r;
    public:
        C(int m1, int n1, int a1, int b1, double x1, double y1,
          double x2, double y2, double z2) : u(), v(a1, b1),
          q(x1, y1), r(x2, y2, z2)
        {
            m = m1 ; n = n1;
        }
};

```

- a. Lớp bao của A và B
- b. Lớp thành phần của A và B
- c. Lớp dẫn xuất của A và B
- d. Lớp cơ sở của A và B

Câu 197. Cho đoạn chương trình như hình trên, các phương thức thao tác trên stack gồm:

```

class stack
{
    private:
        int *data;
        int size;
        int top;
    public:
        stack(int size);
        void push(int x);
        void pop(int &x);
        int isFull();
        int isEmpty();
        ~stack();
};

```

- a. push, pop (1)
- b. isFull, isEmpty, push, pop, Stack, ~Stack
- c. isFull, isEmpty (2)
- d. Cả (1) và (2)

Câu 198. Xét đoạn mã lệnh trên, B sử dụng được các biến thành viên nào của A:

```

class A
{
    int a,b;
    public:
    float F1,F2;
};
class B:public A
{
    ...
}

```

- a. F1, F2
- b. a,b
- c. a,b,F1,F2
- d. Không sử dụng được biến thành viên nào

Câu 199. Cho khai báo thành phần của lớp A như sau, câu nhận định đúng là:

```
class A
{
private:
int a;
public:
A(int b) { a=b; }
};
void main()
{
A x; // Khai báo 1
A y(10); // Khai báo 2
}
```

- a. Cả 2 khai báo hợp lệ.
- b. Lỗi ở dòng (2).
- c. Lỗi ở cả hai dòng khai báo.
- d. Lỗi ở dòng (1).

Câu 200. Chọn các phương án đúng cho mệnh đề “ Trong các phương thức của lớp dẫn xuất, có thể truy nhập trực tiếp tới:”

- a. Các thuộc tính của các đối tượng thành phần.
- b. Các thuộc tính trong chính phương thức đó (1)
- c. Các thuộc tính của lớp cơ sở.
- d. Các thuộc tính mới khai báo trong lớp dẫn xuất (2)
- e. Cả (1) và (2)

Câu 201. Giả sử cho các lớp như trong khai báo như trên. Kết quả hiển thị của đoạn chương trình là:

```

class A
{
float x;
public: A(){ x=1.5; }
void funcA() { cout<<x; }
};
class B: private A
{
float y;
public:
B(): A(){ y=2.5; }
void funcA() { A::funcA(); cout<<y; }
};
void main()
{
B *dt2=new B;
dt2->funcA();
}

```

- a. $x = 1.5$ và $y = 2.5$
- b. $x = 1.5$ và y không xác định.
- c. Giá trị của x và y là không xác định.
- d. x không xác định và $y = 2.5$

Câu 202. class A là bạn của class B, class B là bạn của class C thì:

- a. class C có thể truy xuất đến các thành viên private của class A
- b. class B có thể truy xuất đến các thành viên private của class C
- c. class B có thể truy xuất các thành viên của class A
- d. class A có thể truy xuất các thành viên private của class C

Câu 203. Xem xét bài toán nhập vào danh sách sinh viên gồm n sinh viên với những thông tin: Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Lớp và hiển thị thông tin theo ngày sinh tăng dần. Các lớp đối tượng cần xây dựng cho bài toán gồm:

- a. Lớp Sinh viên và lớp danh sách sinh viên
- b. Lớp Sinh viên
- c. Lớp danh sách sinh viên

Câu 204. Xem xét bài toán nhập vào danh sách sinh viên gồm n sinh viên với những thông tin: Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Lớp và hiển thị thông tin theo ngày sinh tăng dần. Các lớp đối tượng cần xây dựng cho bài toán gồm Lớp Sinh viên và lớp danh sách sinh viên. Các thuộc tính của lớp danh sách sinh viên là:

- a. Số sinh viên (n), mảng đối tượng Sinh viên.
- b. Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Lớp
- c. Số sinh viên (n), Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Lớp

Câu 205. Xem xét bài toán nhập vào danh sách sinh viên gồm n sinh viên với những thông tin: Họ và tên, Ngày sinh (dd/mm/yyyy), Giới tính, Địa chỉ, Lớp và hiển thị thông tin theo ngày sinh tăng dần. Nếu bài toán yêu cầu xây dựng trong C++, các lớp đối tượng cần xây dựng cho bài toán gồm Lớp Sinh viên và lớp danh sách sinh viên. Các thuộc tính của lớp danh sách sinh viên là:

- a. Ngày tháng, Số sinh viên (n), mảng đối tượng Sinh viên.
- b. Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Lớp
- c. Ngày tháng, Số sinh viên (n), Họ và tên, Ngày sinh, Giới tính, Địa chỉ, Lớp
- d. Số sinh viên (n), mảng đối tượng Sinh viên

Câu 206. Cho biết kết quả hiển thị của đoạn chương trình sau?

```
class A
{
    int a,b;
public:
    A() { a=5; b=10; }
    static void hien()
    { cout<<"a="<<a<<"b="<<b; }
};
static A::b=15
void main()
{
    A a;
    a.hien();
}
```

- a. a=5;b=0;
- b. Chương trình có lỗi, không thể hiển thị kết quả.
- c. a=5;b=15;
- d. a=5;b=10;

Câu 207. Chọn phát biểu đúng trong các phát biểu sau:

- a. Mỗi đối tượng sau khi khai báo sẽ được cấp phát một vùng nhớ riêng để chứa các thuộc tính của chúng.
- b. Một lớp (sau khi định nghĩa) có thể xem như một kiểu đối tượng và có thể dùng để khai báo các biến, mảng đối tượng.
- c. Thuộc tính của lớp có thể có kiểu của chính lớp đó.
- d. Tất cả đều đúng.

Câu 208. Khi đa năng hóa toán tử nhập (trích dòng) cho lớp SP ta khai báo dòng tiêu đề như sau:

- a. friend istream & operator >>(istream & istr, SP &a);
- b. friend void istream & operator >>(istream & istr, SP &a);
- c. friend istream & operator >>(istream istr, SP &a);
- d. friend istream & operator >>(istream & istr, SP a);

Câu 209. Cho các lớp sau:

```
class Thi_sinh
{
int SBD;
char Hoten[40];
int Gt;
....
};
class Ds_thi_sinh
{
Thi_sinh *ds;
int sothisinh;

...
};
```

Quan hệ giữa lớp Ds_thi_sinh và Thi_sinh là:

- a. Lớp Thi_sinh là lớp thành phần của lớp Ds_thi_sinh
- b. Lớp Thi_sinh là lớp bao của lớp Ds_thi_sinh
- c. Hai lớp không có mối quan hệ đặc biệt.
- d. Lớp Thi_sinh là lớp con của lớp Ds_thi_sinh

Câu 210. Cho đoạn chương trình như hình trên, Đoạn chương trình sai ở dòng:


```

class A
{
int a;
public:
A(int k) { a=k; }
};
void main()
{
int n=2;
A *a,*b;
a=new A(10); // dòng 1
b=new A[n]; // dòng 2
}

```

- a. Cả hai dòng đều đúng.
- b. Dòng 1 sai.
- c. Cả hai dòng đều sai.
- d. Dòng 2 sai.

Câu 211. Cho đoạn chương trình như hình trên. Kết quả hiển thị của đoạn chương trình là:

```

class A
{
int a;
public:
A() { a=5;}
A(int k)
{ a=k; }
void hien()
{ cout<<"a="<<a;}
};
void main()
{
A a[3]={A(10),A(20)};
a[2].hien();
}

```

- a. a=10;
- b. a=5;

c. a=20;

d. Chương trình có lỗi, không thể hiển thị kết quả.

Câu 212. Đoạn chương trình trên sẽ in ra giá trị là:

```
class A
{
public:
static int n;
};
int A::n=10;
void main()
{
A a;
cout<<++a.n;
}
```

a. Lỗi, không in được a.n.

b. 12.

c. 11.

d. 10.

Câu 213. Cho khai báo thành phần của lớp A như trên, kết quả hiển thị khi chạy đoạn chương trình là:

```
class G
{
protected:
int a,b;
public:
void hien() { cout<<"a="<<a<<"b="<<b;}
G(){x=y=10;}
G(int k1,int k2=8) { a=k1; b=k2;}
};
G a(5);
G b(a) ;
b.hien();
```

a. x=10; y=8;

b. x=10; y=5;

c. x=10; y=10;

d. x=5; y=8;

Câu 214. Hàm mà bất kỳ lớp nào cũng có là:

a. Contructor

b. Hàm friend

c. Không có hàm nào cả

d. Hàm ảo virtual

Câu 215. Chọn phương án tương ứng với phát biểu sai:

a. Một lớp luôn luôn có hàm destructor

b. Hàm destructor có thể là một hàm ảo

c. Hàm destructor dùng để hủy vùng nhớ đã cấp cho con trỏ this

d. Các đáp án trên đều không đúng

Câu 216. Cho khai báo thành phần của lớp A như sau, chọn câu trả lời đúng:

```
class A
{
    int x;
    A(){ x=0; }
};
void main()
{
    A a; // dòng 1
    A *b; // dòng 2
    A *c[10]; // dòng 3
```

a. Lỗi ở dòng 3.

b. Cả 3 khai báo là hợp lệ.

c. Lỗi ở dòng 1.

d. Lỗi ở cả 3 dòng.

Câu 217. Người ta cần quản lý các thông tin về công dân để xác định xem hai công dân bất kỳ có thể kết hôn được với nhau hay không. Biết rằng điều kiện kết hôn là: là công dân từ 18 tuổi trở nên, có tình trạng kết hôn là độc thân, hai công dân kết hôn không cùng giới tính, đồng thời phải lưu lại các thông tin của người kết hôn. Các lớp cần thuộc tính cần xây dựng cho hai lớp đối tượng của bài toán trên là:

- a. Lớp đối tượng cần xây dựng là Lớp Công dân; Các thuộc tính là: họ và tên, ngày sinh, giới tính, tình trạng hôn nhân;
- b. Lớp đối tượng cần xây dựng là Lớp Công dân nam, Công dân nữ; Các thuộc tính là: họ và tên, ngày sinh, tình trạng hôn nhân;
- c. Lớp đối tượng cần xây dựng là Lớp Công dân; Các thuộc tính là: họ và tên, ngày sinh, giới tính, tình trạng hôn nhân; điều kiện kết hôn.
- d. Lớp đối tượng cần xây dựng là Lớp Công dân nam, Lớp kết hôn; Các thuộc tính là: họ và tên, ngày sinh, giới tính, tình trạng hôn nhân;

Câu 218. Trong khai báo dưới:

```
class Diem (1)
{(2)
private: int x,y;(3)
public: (4)
void Diem(int x,int y);(5)
void ~Diem();(6)
};(7)
```

Sai tại dòng 5 và 6 vì:

- a. Hàm tạo và hàm hủy của lớp không bắt đầu bằng từ khóa void.
- b. Chỉ được phép xây dựng hàm tạo hoặc hàm hủy trong một lớp.
- c. Hàm hủy không được bắt đầu bằng từ khóa void
- d. Hàm hủy phải có tham đối truyền vào, còn hàm tạo thì không.

Câu 219. Cho đoạn chương trình sau:

```
class A{
private:
int x,y;
protected:
void Nhap();
};
```

Lớp B kế thừa lớp A, để phương thức Nhap của lớp A trở thành thành phần protected của lớp B ta chọn loại kế thừa là:

- a. Private.
- b. Public.
- c. Protected.
- d. Private hoặc Protected.

Câu 220. Cho đoạn chương trình sau:

```
class A{
private:
int x,y;
protected:
```

```
void Nhap();  
};
```

Lớp B kế thừa lớp A, để phương thức Nhap của lớp A trở thành thành phần public của lớp B ta chọn loại kế thừa là:

- a. Không có dạng kế thừa nào phù hợp.
- b. Public.
- c. Protected.
- d. Private hoặc Protected.

Câu 221 Cho 2 lớp như hình trên. Khi đó, nếu MyA là một đối tượng lớp A, muốn thực hiện phương thức show của lớp B thì câu lệnh chấp nhận được là câu lệnh:

```
class B  
{  
public:  
void show();  
};  
class A : public B  
{  
public:  
void show();  
};
```

- a. MyA.B.show();
- b. MyA.B::show();
- c. B::MyA.show();
- d. MyA.show();

Câu 222. Cho khai báo lớp như hình trên. Giả sử trong hàm main có khai báo đối tượng p như sau: A p; Câu lệnh đúng khi viết trong hàm main là:

```

class A
{
private:
int x;
protected:
float y;
public:
char z;
void hien_thi()
{cout<<"Gia tri A la"<<x<<" "<<y<<" "<<z;}
};

```

- a. cin>>p.z;
- b. cin>>p.y;
- c. cả ba đáp án đều đúng
- d. cin>>p.x;

Câu 223. Cho biết khẳng định đúng nhất trong các khẳng định sau:

- a. Một lớp A có thể là bạn của nhiều lớp.
- b. Nếu lớp A là bạn của lớp B thì chỉ có một số phương thức của A có thể truy xuất đến các thành phần riêng của lớp B.
- c. Cả ba phương án đều đúng.
- d. Nếu lớp A là bạn của lớp B thì lớp B cũng là bạn của lớp A.

Câu 224. Cho khai báo thành phần của các lớp như trên, Có lỗi trong phương thức nhap() của lớp HCN. Lỗi đó ở dòng:

```

class POINT
{
int x,y;
public:
void nhap();
};
class HCN
{
POINT p;


---


int d,r;
public:
void nhap()
{
cout<<"d,r="; cin>>d,r; // dòng 1
cout<<"x,y="; cin>>p.x,p.y; // dòng 2
}
};

```

- Lỗi ở dòng 1.
- Lỗi ở dòng 2.
- Lỗi ở cả 2 dòng.
- Không dòng nào lỗi.

Câu 225. Cho đoạn chương trình như hình trên. Đoạn chương trình bị sai tại dòng:

```

class SUN
{
protected:
int x;
public:
SUN(int k = 5) { x = k; }
};
void main( )
{
int n=2;
SUN *a,*b;
a = new SUN(10); // dòng 1
b = new SUN(n); // dòng 2
}

```

- a. Dòng 1 sai.
- b. Dòng 2 sai.
- c. Cả hai dòng đều sai.
- d. Cả hai dòng đều đúng.

Câu 226. Đoạn chương trình trên xây dựng template tìm giá trị lớn nhất. Việc sử dụng template này để xác định giá trị lớn nhất của hai biến ten1, ten2 là:


```

#include <conio.h>
#include <iostream.h>
template <class T> T giatrimax(T a, T b)
{
    if(a>b)
        return a;
    else
        return b;
}
void main()
{
    int x=10, y=4;
    float z=9.3, t=-2.3;
    char *ten1= "Hang", *ten2= "Thanh";
    cout<< "\n Gia tri lon nhat cua x va y la:"<<giatrimax(x,y);
    cout<< "\n Gia tri lon nhat cua z va t la:"<<giatrimax(z,t);
    cout<< "\n Gia tri lon nhat cua ten1 va ten2 la:"<<giatrimax(ten1,ten2);
    getch();
}

```

- Không thể sử dụng được.
- Hoàn toàn có thể sử dụng được.
- Cần xác định kiểu dữ liệu ten1, ten2 là kiểu class T

Câu 227. Câu lệnh khai báo như sau:

```

template (class T) T min (T a, T b)
{
    ...
}

```

có nghĩa là:

- Khai báo họ các hàm tìm giá trị nhỏ nhất giữa hai số có cùng kiểu dữ liệu.
- Khai báo họ các hàm tìm giá trị nhỏ nhất giữa các cố bất kỳ.
- Khai báo họ các hàm tìm giá trị nhỏ nhất của một dãy các số có chung kiểu dữ liệu

Câu 228. Câu lệnh khai báo như sau:

```

template (class T) T min (T a, T b, T c)
{
    ...
}

```

```
}
```

có nghĩa là:

- a. Khai báo họ các hàm tìm giá trị nhỏ nhất giữa hai số có cùng kiểu dữ liệu.
- b. Khai báo họ các hàm tìm giá trị nhỏ nhất giữa các số bất kỳ.
- c. Khai báo họ các hàm tìm giá trị nhỏ nhất của một dãy các số có chung kiểu dữ liệu
- d. Khai báo họ các hàm tìm giá trị nhỏ nhất trong ba số có cùng kiểu dữ liệu

Câu 229. Câu lệnh khai báo như sau:

```
template (class T, class U) T min (T a, U b)
```

```
{
```

```
...
```

```
}
```

có nghĩa là:

- a. Khai báo họ các hàm tìm giá trị nhỏ nhất giữa hai số có cùng kiểu dữ liệu.
- b. Khai báo họ các hàm tìm giá trị nhỏ nhất giữa hai số khác kiểu dữ liệu.
- c. Khai báo họ các hàm tìm giá trị nhỏ nhất của một dãy các số có chung kiểu dữ liệu
- d. Khai báo họ các hàm tìm giá trị nhỏ nhất trong hai số có thể cùng hay khác kiểu dữ liệu

Câu 230. Cho khai báo như hình trên, khai báo là:

```
template <class T> class Diem
```

```
{
```

```
    T x,y;
```

```
    public:
```

```
    Diem (T x=0, T y=0);
```

```
    void hienthi();
```

```
};
```

- a. Khai báo khuôn hình hàm.
- b. Khai báo khuôn hình lớp.
- c. Khai báo một tệp template tạm thời

Câu 231. Cho khai báo như trên, câu nhận định đúng là:

```
template <class T> class Diem
{
    T x,y;
    public:
    Diem (T x=0, T y=0);
    void hienthi();
};
```

- a. Khai báo khuôn hình lớp Điểm với kiểu dữ liệu thuộc tính bất kỳ.
- b. Khai báo khuôn hình lớp T với thuộc tính là một điểm bất kỳ.
- c. Khai báo khuôn hình lớp Điểm không xác định kiểu dữ liệu của thuộc tính.