

Câu 1: Thuật ngữ nào sau đây không phải là đặc trưng của lập trình hướng đối tượng?	Đáp án đúng: Exceptions
Câu 2: Đặc trưng nào sau đây được xem như là sự che dấu dữ liệu (data hiding)?	Đáp án đúng: Encapsulation
Câu 3: Từ nào sau đây KHÔNG phải là từ khóa trong C++?	Đáp án đúng: extends
Câu 4: Một lớp (class) được xem như là khung mẫu (template) của	Đáp án đúng: Đối tượng (object)
Câu 5: Tính chất nào cho phép lớp kế thừa thuộc tính hoặc phương thức của lớp khác?	Đáp án đúng: Thừa kế
Câu 6: Các phương thức của cùng một lớp có cùng tên, khác số tham số và kiểu tham số được gọi là gì?	Đáp án đúng: Overloading
Câu 7: Khi các phương thức được định nghĩa trong một lớp có thể được gọi bởi các đối tượng thuộc lớp đó. Điều này được gọi là:	Đáp án đúng: Gửi thông điệp cho đối tượng
Câu 8: Phương pháp lập trình theo lối tuần tự được gọi là phương pháp lập trình:	Đáp án đúng: Tuyến tính
Câu 9: Trong C++, khi một đối tượng được tạo ra bởi từ khóa new sẽ được lưu ở đâu trong bộ nhớ?	Đáp án đúng: heap
Câu 10: Lớp cơ sở trừu tượng (abstract class) có thể tạo ra bao nhiêu đối tượng?	Đáp án đúng: 0
Câu 11: Ngôn ngữ nào sau đây không hỗ trợ lập trình hướng đối tượng?	Đáp án đúng: Pascal
Câu 12: Phát biểu nào sau đây là đúng về sự đóng gói?	Đáp án đúng: Là cơ chế ràng buộc giữa dữ liệu và thao tác trên dữ liệu đó thành một thể thống nhất

Câu 13: Phương pháp thiết kế trên xuống (top-down) thường được sử dụng trong phương pháp lập trình nào?	Đáp án đúng: Có cấu trúc
Câu 14: Các thành phần chung của đối tượng cung cấp một _____ điều khiển các thành phần riêng của đối tượng.	Đáp án đúng: giao diện
Câu 15: Trong C++, khai báo prototype hàm nào sau đây là sai?	Đáp án đúng: int Area(int a=0,int b);
Câu 16: Trong C++, để cấp phát bộ nhớ động ta dùng toán tử:	Đáp án đúng: new
Câu 17: Quan hệ _____ cho phép một lớp có thể truy xuất phương thức hay thuộc tính của lớp khác.	Đáp án đúng: friend
Câu 18: Phạm vi truy xuất nào không có trong các lớp của C++?	Đáp án đúng: friend
Câu 19: Hàm destructor trả về giá trị gì?	Đáp án đúng: Không trả về giá trị nào cả
Câu 20: Hàm mà bất kỳ lớp (class) nào cũng có là hàm gì?	Đáp án đúng: Hàm xây dựng và hàm hủy
Câu 21: Khi khai báo lớp trong C++, hành vi của đối tượng được khai báo như là:	Đáp án đúng: Hàm hay phương thức
Câu 22: Có bao nhiêu phạm vi truy xuất trong C++?	Đáp án đúng: 3
Câu 23: Constructor được sử dụng để làm gì?	Đáp án đúng: Khởi tạo đối tượng và khởi tạo dữ liệu thành viên
Câu 24: Hàm nào được gọi khi một đối tượng được tạo ra?	Đáp án đúng: Hàm khởi tạo
Câu 25: Để xóa đối tượng ra khỏi bộ nhớ, hàm nào sau đây sẽ được gọi?	Đáp án đúng: Hàm hủy
Câu 26: Từ khóa nào được sử dụng để định nghĩa một lớp trong C++?	Đáp án đúng: class

Câu 27: Câu lệnh nào sau đây được sử dụng để lớp A khai báo lớp B là bạn của nó:	Đáp án đúng: friend class B;
Câu 28: _____ được xây dựng để tạo ra các đối tượng có cùng kiểu (type)	Đáp án đúng: Lớp
Câu 29: Hàm khởi tạo mặc định là hàm cùng tên với tên lớp và _____.	Đáp án đúng: Có 0 tham số
Câu 30: Để khai báo dữ liệu thành viên tĩnh, ta sử dụng từ khóa	Đáp án đúng: static
Câu 31: Hàm thành viên tĩnh có thể truy cập đến thành phần dữ liệu được khai báo	Đáp án đúng: static
Câu 32: Thông thường, dữ liệu thành viên của một lớp phải được khai báo	Đáp án đúng: private
Câu 33: Một đối tượng là thể hiện của	Đáp án đúng: lớp
Câu 34: Giả sử ta có con trỏ p thuộc lớp P, để truy cập hàm show() của lớp P câu lệnh nào sau đây là đúng?	Đáp án đúng: p->show();
Câu 35: Phát biểu nào sau đây là sai về tính đóng gói (encapsulation) trong lập trình hướng đối tượng?	Đáp án đúng: Đóng gói dữ liệu làm cho chương trình ngắn gọn
Câu 36: Giả sử ta có lớp Circle với một thuộc tính là radius (kiểu số thực). Khai báo hàm khởi tạo sao chép nào sau đây là đúng?	Đáp án đúng: Circle (const Circle&)
Câu 37: Khi ra khỏi một phạm vi (block), đối tượng thuộc phạm vi đó sẽ	Đáp án đúng: Được hủy (xóa khỏi bộ nhớ)
Câu 38: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include<iostream> using namespace std; class MyClass{ private: int x; public: void Print() { cout<<x; } }; int main(){ MyClass x; cout<<x::x; return 0; }	Đáp án đúng: Lỗi

Câu 39: Để hàm tự do có thể truy cập thành phần dữ liệu private của một lớp, phát biểu nào sau đây là đúng?	Đáp án đúng: Hàm tự do phải là bạn của lớp
Câu 40: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream> using namespace std; class MyClass{ private: int x; public: void Print() { cout<<x; } }; int main(){ MyClass *x; x.Print() ; return 0; }</code>	Đáp án đúng: Lỗi
Câu 41: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream> using namespace std; class MyClass{ private: int x; public: void Print() { cout<<x; } }; int main(){ MyClass x; x.Print(); return 0; }</code>	Đáp án đúng: Số nguyên ngẫu nhiên
Câu 42: Kết quả biên dịch, thực thi đoạn chương trình sau: <code>#include<iostream> using namespace std; class Test {};</code> void main() {Test t;}	Đáp án đúng: Chương trình thực thi mà không xuất ra gì hết.
Câu 43: Phát biểu nào sau đây là SAI về hàm xây dựng?	Đáp án đúng: Hàm có thể trả về một giá trị.
Câu 44: Nếu lớp Box có hàm khởi tạo là <code>Box(int=0,int=0)</code> , để khai báo đối tượng thuộc lớp Box ta sử dụng câu lệnh:	Đáp án đúng: <code>Box b;</code>
Câu 45: Để một hàm có thể truy cập các thuộc tính thuộc thành phần private của lớp A thì hàm này phải là	Đáp án đúng: Hàm bạn của lớp A
Câu 46: Câu lệnh nào sau đây được sử dụng để khai báo và cấp phát bộ nhớ cho con trỏ thuộc lớp Line?	Đáp án đúng: <code>Line *a=new Line();</code>
Câu 47: Giả sử ta muốn xây dựng hàm dùng để truy cập thuộc tính gioitinh có kiểu logic của một lớp. Khai báo hàm nào sau đây là hợp lý?	Đáp án đúng: <code>bool getGioitinh()</code>
Câu 48: Từ khoá protected trong một lớp có ý nghĩa gì?	Đáp án đúng: Khai báo các thành viên của lớp chỉ được thừa kế

Câu 49: Con trỏ this được sử dụng để tham chiếu đến:	Đáp án đúng: Dữ liệu thành viên và hàm thành viên của một đối tượng
Câu 50: Giả sử lớp A là bạn của lớp B, lớp B là bạn của lớp C, phát biểu nào sau đây là SAI?	Đáp án đúng: Lớp A có thể truy cập dữ liệu thành viên của lớp C
Câu 51: Khi thực thi đoạn chương trình sau kết quả sẽ là: <code>#include<iostream> using namespace std; class B { public: B() { }; int k; }; int main() { B b; cout << b.k << endl; return 0; }</code>	Đáp án đúng: Xuất ra màn hình số nguyên bất kỳ
Câu 52: Kết quả biên dịch, thực thi chương trình sau: <code>#include<iostream> using namespace std; class Point { int n; public: Point(int x) { n = x; } void Print(){cout<<n; } }; int main() { Point pt(4); pt.Print(); return 0; }</code>	Đáp án đúng: Chương trình thực thi xuất ra màn hình: 4
Câu 53: Hàm nào trong đoạn chương trình sau đây được định nghĩa bên trong lớp A? <code>class A { public: A() { value = 0; } A(double); double f1() { return value; } double f2(); private: double value; };</code>	Đáp án đúng: A(), f1()
Câu 54: Giả sử ta có khai báo: <code>Car *pcar;</code> câu lệnh nào sau đây được sử dụng để cấp phát bộ nhớ cho pcar?	Đáp án đúng: <code>pcar=new Car[10];</code>
Câu 55: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream> using namespace std; class Test { public: int x; Test() { cout <<"Test"; } }; int main() { Test test; cout << test.x; return 0; }</code>	Đáp án đúng: Xuất ra màn hình số nguyên bất kỳ
Câu 56: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream> using namespace std; class A { public: int x; int y; int z; A(): x(1), y(2), z(3) { } }; int main() { A a; cout << a.x <<" "<< a.y <<" "<< a.z; return 0; }</code>	Đáp án đúng: 1 2 3

Câu 57: Khi thực thi đoạn chương trình sau kết quả sẽ là gì? <code>#include<iostream> using namespace std; class Teacher { public: static int n; public : Teacher () { cout<<" "<<n++ ; } }; int Teacher::n = 0; void main() { Teacher t1; Teacher t2; Teacher t3; cout<<" "<<t1.n ; getch(); }</code>	Đáp án đúng: Xuất ra màn hình: 0 1 2 3
Câu 58: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include <iostream> using namespace std; class rect { int x, y; public: void val (int, int); int area () { return (x * y); } }; void rect::val (int a, int b) { x = a; y = b; } int main () { rect rect; rect.val (3, 4); cout << "rect area: " << rect.area(); return 0; }</code>	Đáp án đúng: rect area: 12
Câu 59: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream> using namespace std; class Foo { public: int x; int y; Foo() { x =10; y =10; } void p() { int x =20; // local variable cout <<"x is " << x <<" "; cout <<"y is " << y << endl; } }; int main() { Foo foo; foo.p(); return 0; }</code>	Đáp án đúng: x is 20 y is 10
Câu 60: Khi thực thi đoạn chương trình sau kết quả sẽ là gì? <code>#include<iostream> using namespace std; class Employee { char ten[30]; char ms[10]; int tuoi; public : Employee (char te[], char m[], int tu):tuoi(tu) { strcpy(ten, te); strcpy(ms, m); } void Display(){ cout<<"Ma so: " << ms <<" Ten: " << ten <<" Tuổi:" << tuoi << endl ; } }; void main() { Employee e(" Nguyen Van A", "001", 20); e.Display(); }</code>	Đáp án đúng: Lỗi biên dịch.
Câu 61: Giả sử ta có lớp Rational (phân số), để tái định nghĩa toán tử gán thì định nghĩa nào sau đây là đúng?	Đáp án đúng: <code>Rational operator=(const Rational& secondRational);</code>
Câu 62: Giả sử ta có lớp Rational (phân số), để tái định nghĩa toán tử hậu tố ++, chỉ thị nào sau đây là đúng?	Đáp án đúng: <code>Rational operator++(int dummy)</code>

Câu 63: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <code>#include <iostream> using namespace std; int main() { int a; a = 5 + 3 * 5; cout << a; return 0; }</code>	Đáp án đúng: 20
Câu 64: Toán tử nào sau đây KHÔNG được phép quá tải?	Đáp án đúng: ::
Câu 65: Toán tử nào sau đây không được phép quá tải?	Đáp án đúng: :: , . , .* , ?:
Câu 66: Toán tử được quá tải phải được định nghĩa như: 1. Static member functions 2. Non- static member functions 3. Friend Functions	Đáp án đúng: 2 và 3
Câu 67: Khai báo nào sau đây là đúng khi quá tải toán tử cộng?	Đáp án đúng: operator+
Câu 68: Để quá tải toán tử += ta cần:	Đáp án đúng: Quá tải toán tử +=
Câu 69: Toán tử nào sau đây được phép quá tải?	Đáp án đúng: !=
Câu 70: Giả sử ta có lớp Rational (phân số), định nghĩa hàm nào sau đây được sử dụng để chuyển một Rational (phân số) sang kiểu số thực?	Đáp án đúng: operator double()
Câu 71: Giả sử ta có lớp Rational (phân số), để tái định nghĩa toán tử + ta sử dụng khai báo nào?	Đáp án đúng: Rational Rational::operator+(const Rational& r)
Câu 72: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <code>#include <iostream> using namespace std; int main() { int i, j; j = 10; i = (j++, j + 100, 999 + j); cout << i; return 0; }</code>	Đáp án đúng: 1010

Câu 73: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <pre>#include<iostream> using namespace std; class Point { private: int x, y; public: Point() : x(0), y(0) { } Point& operator()(int dx, int dy); void show() { cout << "x = " << x << ", y = " << y; } }; Point& Point::operator()(int dx, int dy) { x = dx; y = dy; return *this; } int main() { Point pt; pt(3, 2); pt.show(); return 0; }</pre>	Đáp án đúng: x = 3, y = 2
Câu 74: Toán tử nào sau đây thường được quá tải như hàm toàn cục?	Đáp án đúng: Toán tử <<
Câu 75: Toán tử nào sau đây thường được quá tải như là hàm bạn?	Đáp án đúng: >>, <<
Câu 76: Giả sử y và z là hai đối tượng thuộc lớp Data với toán tử += đã được quá tải. Câu lệnh nào sau đây tương đương với câu lệnh y += z?	Đáp án đúng: y.operator+=(z)
Câu 77: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <pre>#include<iostream> using namespace std; class A { int i; public: A(int ii = 0) : i(ii) { } void show() { cout << i <<" " } }; class B { int x; public: B(int xx) : x(xx) { } operator A() const { return A(x); } }; void g(A a) { a.show(); } int main() { B b(10); g(b); g(20); return 0; }</pre>	Đáp án đúng: Lỗi
Câu 78: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: #include <iostream> using namespace std; class Test2 { int y; }; class Test { int x; Test2 t2; public: operator Test2 () { return t2; } operator int () { return x; } }; void fun (int x) { cout << "fun(int) called"<<endl; } void fun (Test2 t) { cout << "fun(Test 2) called"<<endl; } int main() { Test t; fun(t); return 0; }	Đáp án đúng: Lỗi

Câu 79: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <code>#include <iostream> #include <cmath> using namespace std; class Complex { private: double real; double imag; public: Complex(double r = 0.0, double i = 0.0) : real(r), imag(i) {} double mag() { return getMag(); } operator double () { return getMag(); } private: double getMag() { return sqrt(real * real + imag * imag); } }; int main() { Complex com(3.0, 4.0); cout << com.mag(); cout << com; return 0; }</code>	Đáp án đúng: 55
Câu 80: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <code>#include <iostream> using namespace std; class sample { public: sample(int i) : m_i(i) {} public: int operator()(int i = 0) const { return m_i + i; } operator int () const { return m_i; } private: int m_i; friend int g(const sample&); }; int f(char c) { return c; } int main() { sample f(2); cout << f(2); return 0; }</code>	Đáp án đúng: 4
Câu 81: Đơn thừa kế là gì?	Đáp án đúng: Một lớp kế thừa từ một lớp khác.
Câu 82: Phát biểu nào sau đây là đúng về đa thừa kế?	Đáp án đúng: Một lớp kế thừa từ hai lớp trở lên
Câu 83: Giả sử lớp A thừa kế từ lớp B, cả A và B đều không có hàm khởi tạo mặc định. Để gọi hàm khởi tạo mặc định của lớp B từ A ta sử dụng câu lệnh:	Đáp án đúng: A(): B() { ... }
Câu 84: Phát biểu nào sau đây là đúng về lớp cơ sở trừu tượng (abstract class)?	Đáp án đúng: Là lớp không thể tạo ra thể hiện của nó
Câu 85: Hàm thuần ảo (pure virtual function) là hàm	Đáp án đúng: Phải được định nghĩa lại ở lớp dẫn xuất (lớp con)
Câu 86: Giả sử lớp GeometricObject thừa kế từ lớp Circle và p là con trỏ thuộc lớp GeometricObject. Để ép kiểu p sang Circle ta sử dụng câu lệnh	Đáp án đúng: Circle* p1 = dynamic_cast<Circle*>(p);
Câu 87: Hàm nào sau đây là hàm thuần ảo?	Đáp án đúng: virtual double getArea() = 0;

Câu 88: Cho đoạn mã lệnh như bên dưới, đối tượng bất kỳ thuộc lớp Y có thể truy xuất phương thức nào? class X { private: int M; void Func2(); public: X(int iM = 0); void Func1(); ~X(); }; class Y : public X { private: float F; public: Y(); Y(float iF, int iM); void Func3(); ~Y(); };	Đáp án đúng: Func1() và Func3()
Câu 89: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include<iostream> using namespace std; class B { public: ~B() { cout << "B"; } }; class A: public B { public: ~A() { cout << "A"; } }; int main() { A a; return 0; }	Đáp án đúng: AB
Câu 90: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include<iostream> using namespace std; class B { public: B() { cout << "B"; } }; class A: public B { public: A() { cout << "A"; } }; int main() { A a; return 0; }	Đáp án đúng: BA
Câu 91: Lỗi của đoạn chương trình sau là gì? class Fruit { public: Fruit(int id) { } }; class Apple: public Fruit { public: Apple() { } };	Đáp án đúng: Định nghĩa hàm khởi tạo của lớp Apple không đúng
Câu 92: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include<iostream> using namespace std; class base { int arr[10]; }; class b1: public base { }; class b2: public base { }; class derived: public b1, public b2 { }; int main(void) { cout << sizeof(derived); return 0; }	Đáp án đúng: 80
Câu 93: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include<iostream> using namespace std; class Base { public: int fun() { cout << "Base::fun() called"; } int fun(int i) { cout << "Base::fun(int i) called"; } }; class Derived: public Base { public: int fun() { cout << "Derived::fun() called"; } }; int main() { Derived d; d.fun(5); return 0; }	Đáp án đúng: Lỗi biên dịch

Câu 94: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream></code> <code>#include<string> usingnamespacestd;</code> <code>classC { public: string toString() { return"C"; } }; classB:publicC { string toString() { return"B"; } }; classA:publicB { string toString() { return"A"; } }; voiddisplayObject(C* p) { cout << p->toString(); } intmain() { A a; B b; C c;</code> <code>displayObject(&a); displayObject(&a); displayObject(&b); displayObject(&c); return0; }</code>	Đáp án đúng: CCC
Câu 95: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream></code> <code>#include<string> usingnamespacestd;</code> <code>classPerson { public: voidprintInfo() { cout << getInfo() << endl; } string getInfo() { return"Person"; } }; classStudent:publicPerson { public: string getInfo() { return"Student"; } }; intmain() { Person().printInfo(); Student().printInfo(); return 0; }</code>	Đáp án đúng: Person Person
Câu 96: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream></code> <code>#include<string> usingnamespacestd;</code> <code>classA { public: string toString() { return"A"; } }; classB:publicA { public: string toString() { return"B"; } }; intmain() { B b; cout << static_cast<A>(b).toString() << b.toString() << endl; return0; }</code>	Đáp án đúng: AB
Câu 97: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream></code> <code>usingnamespacestd; classA { public: A() { t(); cout <<"i from A is " << i << endl; } voidt() { setI(20); } virtualvoidsetI(inti) { this->i =2* i; } inti; }; classB:publicA { public: B() { // cout << "i from B is " << i << endl; } virtualvoidsetI(inti) { this->i =3* i; } }; intmain() { A* p =newB(); return0; }</code>	Đáp án đúng: Hiển thị chuỗi: "i from A is 40".

Câu 98: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <pre>#include<iostream> using namespace std; class Base { public: Base() { cout<<"Constructor: Base"<<endl; } virtual ~Base() { cout<<"Destructor : Base"<<endl; } }; class Derived: public Base { public: Derived() { cout<<"Constructor: Derived"<<endl; } ~Derived() { cout<<"Destructor : Derived"<<endl; } }; int main() { Base *Var = new Derived(); delete Var; return 0; }</pre>	Đáp án đúng: Constructor: Base Constructor: Derived Destructor : Derived Destructor : Base
Câu 99: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include<iostream> using namespace std; class Base { public: virtual void show() { cout<<" In Base "; } }; class Derived: public Base { public: void show() { cout<<"In Derived "; } }; int main(void) { Base *bp = new Derived; bp->Base::show(); return 0; }	Đáp án đúng: In Base
Câu 100: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include <iostream> using namespace std; class A { public: virtual void fun(); }; class B { public: void fun(); }; int main() { int a = sizeof(A), b = sizeof(B); if (a == b) cout << "a == b"; else if (a > b) cout << "a > b"; else cout << "a < b"; return 0; }	Đáp án đúng: a > b
Câu 101: OOP được viết tắt từ cụm từ nào?	Đáp án đúng: Object-Oriented Programming
Câu 102: Đối tượng (object) được xem như như thể nào của lớp (class)?	Đáp án đúng: Thể hiện
Câu 103: Một đối tượng là thể hiện cụ thể của	Đáp án đúng: Lớp
Câu 104: Một lớp định nghĩa lại phương thức của lớp cơ sở của nó được gọi là	Đáp án đúng: Override
Câu 105: C++ cho phép đa năng hóa:	Đáp án đúng: Toán tử và hàm

Câu 106: Một trong những ưu điểm của phương pháp lập trình hướng đối tượng so với các phương pháp lập trình truyền thống là	Đáp án đúng: Khả năng tái sử dụng lại lớp
Câu 107: Ưu điểm nào sau đây không phải là ưu điểm của lập trình có cấu trúc?	Đáp án đúng: Phù hợp với các phần mềm lớn
Câu 108: Giả sử ta có prototype của 3 hàm sau: int MyAbs(int i); //(1) long MyAbs (long l); //(2) double MyAbs (double d); //(3) Để gọi hàm (2) ta sử dụng câu lệnh nào?	Đáp án đúng: MyAbs (10l);
Câu 109: Mục đích xây dựng lớp (class) trong phương pháp lập trình hướng đối tượng là:	Đáp án đúng: Đóng gói dữ liệu, sử dụng lại các lớp, tạo cơ chế mô hình hóa các đối tượng trong thế giới thực. Tất cả đều đúng
Câu 110: Khai báo lớp (class) nào sau đây là hợp lệ?	Đáp án đúng: class A { int x; };
Câu 111: Lớp (class) chứa thành phần nào sau đây:	Đáp án đúng: Dữ liệu thành viên và hàm thành viên
Câu 112: Hàm xây dựng (constructor) không có tham số được gọi là hàm xây dựng gì?	Đáp án đúng: Mặc định
Câu 113: Để xóa đối tượng a thuộc lớp A ta sử dụng câu lệnh nào trong những câu lệnh sau đây?	Đáp án đúng: delete a;
Câu 114: Câu lệnh nào sau đây được sử dụng để lớp A khai báo hàm F() là bạn của nó:	Đáp án đúng: friend F();
Câu 115: Khi khai báo một đối tượng thuộc một lớp đã có, hàm nào của đối tượng sẽ được gọi?	Đáp án đúng: Hàm xây dựng tương ứng
Câu 116: Thành phần dữ liệu public của một lớp có thể truy cập bởi	Đáp án đúng: Hàm tự do, hàm thành viên, hàm bạn
Câu 117: Khi cài đặt phương thức bên ngoài phạm vi, lớp dùng chỉ thị nào?	Đáp án đúng: ::

Câu 118: Giả sử ta có khai báo: Car *pcar; câu lệnh nào sau đây được sử dụng để giải phóng bộ nhớ cho pcar?	Đáp án đúng: delete[]Car;
Câu 119: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include<iostream> usingnamespacestd; class MyClass{ private: int x; public: void Print() { cout<<x; } }; int main(){ MyClass x; cout<<x; return 0; }	Đáp án đúng: Lỗi
Câu 120: Quan hệ “bạn” nào sau đây là SAI?	Đáp án đúng: Lớp là bạn của hàm tự do
Câu 121: Phát biểu nào sau đây là SAI khi nói về lớp?	Đáp án đúng: Lớp được tạo ra bởi đối tượng.
Câu 122: Phát biểu nào sau đây là đúng về hàm hủy (destructor)?	Đáp án đúng: Hàm hủy dùng để xóa đối tượng khỏi bộ nhớ.
Câu 123: Câu lệnh nào sau đây là đúng khi khai báo hàm khởi tạo sao chép cho lớp X?	Đáp án đúng: X(const X &x);
Câu 124: Giả sử ta có box1 và box2 là hai đối tượng thuộc lớp Box. Ý nghĩa của câu lệnh box1=box2 là gì?	Đáp án đúng: Sao chép thuộc tính của box2 vào box1
Câu 125: Thành phần dữ liệu protected của một lớp có thể truy cập bởi	Đáp án đúng: Lớp dẫn xuất
Câu 126: Kết quả biên dịch, thực thi chương trình sau: #include<iostream> usingnamespacestd; class ABC { int n; public: ABC(int x){n=x; } void Print(){ cout<<n; } }; void main() { ABC t; t.Print(); }	Đáp án đúng: Lỗi biên dịch.
Câu 127: Định nghĩa hàm khởi tạo (constructor) nào sau đây là đúng nhất? { ... }	Đáp án đúng: Là hàm có tên trùng với tên lớp, được gọi ngay sau khi khai báo đối tượng hoặc con trỏ đối tượng, có thể kế thừa.

Câu 128: Khi thực thi đoạn chương trình sau kết quả sẽ là gì? <code>#include<iostream> usingnamespacestd; classA { public: ints; A(intnewS) { s = newS; } voidprint() { cout << s; } }; intmain() { A a; a.print(); return 0; }</code>	Đáp án đúng: Lỗi vì lớp A không có hàm khởi tạo mặc định
Câu 129: Khi thực thi đoạn chương trình sau kết quả sẽ là gì? <code>#include<iostream> usingnamespacestd; class AAA { int na; public : AAA(int a=0) { na = a ; } ~AAA(){cout<<" "<<na ;} }; void Func(AAA aaa){ AAA *a1 = new AAA(3); delete a1; } void main() { AAA aaa(4); Func(aaa); }</code>	Đáp án đúng: Xuất ra màn hình: 3 4 4
Câu 130: Lỗi của đoạn chương trình sau là gì? <code>#include<iostream> usingnamespacestd; classTempClass { public: inti; TempClass() { inti =5; } }; intmain() { TempClass temp(2); return 0; }</code>	Đáp án đúng: Lỗi vì lớp TempClass không có hàm khởi tạo có đối số là số nguyên
Câu 131: Toán tử nào sau đây không được phép quá tải (tái định nghĩa)?	Đáp án đúng: ?:
Câu 132: Giả sử ta có lớp Rational (phân số), để tái định nghĩa toán tử tiền tố --, chỉ thị nào sau đây là đúng?	Đáp án đúng: Rational operator--()
Câu 133: Có bao nhiêu toán tử được phép quá tải trong danh sách các toán tử sau đây? <code>::'[]' '->' '.' '*'</code>	Đáp án đúng: 2
Câu 134: Định nghĩa hai hàm thành viên của một lớp với tên giống nhau trong lập trình hướng đối tượng được gọi là:	Đáp án đúng: Overloading
Câu 135: Giả sử tồn tại lớp Data, để quá tải toán tử xuất << thì câu lệnh nào sau đây là đúng?	Đáp án đúng: <code>ostream &operator<<(ostream &output, const Data &dataToPrint)</code>
Câu 136: Giả sử ta có lớp Rational (phân số), để tái định nghĩa toán tử nhập >>, chỉ thị nào sau đây cần được khai báo bên trong lớp Rational?	Đáp án đúng: <code>friend istream& operator>>(istream& stream, Rational& rational);</code>

Câu 137: Toán tử nào sau đây được mặc định quá tải bởi trình biên dịch?	Đáp án đúng: Toán tử =
Câu 138: Hàm quá tải toán tử được gọi khi nào?	Đáp án đúng: Khi sử dụng toán tử
Câu 139: Toán tử nào sau đây phải được quá tải như hàm thành viên của lớp?	Đáp án đúng: Toán tử chuyển (ép) kiểu
Câu 140: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <pre>include<stdlib.h> #include<stdio.h> #include<iostream> using namespace std; class Test { int x; public: void* operator new(size_t size); void operator delete(void*); Test(int i) { x = i; cout << "Constructor called \n"; } ~Test() { cout << "Destructor called \n"; } }; void* Test::operator new(size_t size) { void *storage = malloc(size); cout << "new called \n"; return storage; } void Test::operator delete(void *p) { cout<<"delete called \n"; free(p); } int main() { Test *m = new Test(5); delete m; return 0; }</pre>	Đáp án đúng: new called Constructor called Destructor called delete called
Câu 141: Phương pháp lập trình hướng đối tượng cho phép một lớp kế thừa từ lớp đã có. Đặc tính này được gọi là:	Đáp án đúng: inheritance
Câu 142: Câu lệnh "class A: public B" có nghĩa là:	Đáp án đúng: B là lớp dẫn xuất từ A
Câu 143: Quá trình xây dựng một lớp mới dựa trên lớp đã có được gọi là	Đáp án đúng: Inheritance
Câu 144: Giả sử lớp Diem2D kế thừa từ lớp Diem. Để gọi hàm toString() của lớp Diem từ đối tượng d thuộc lớp Diem2D ta sử dụng câu lệnh nào trong các lệnh sau đây?	Đáp án đúng: d.Diem::toString();

Câu 145: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream></code> <code>using namespace std; class ParentClass { public: int id; ParentClass(int id) { this->id = id; } void print() { cout << id << endl; } }; class ChildClass: public ParentClass { public: int id; ChildClass(int id): ParentClass(1) { this->id = id; } }; int main() { ChildClass c(2); c.print(); return 0; }</code>	Đáp án đúng: 1
Câu 146: Khi thực thi đoạn chương trình sau kết quả trên màn hình sẽ là: <code>#include<iostream> using namespace std; class Base { public: virtual void show() = 0; }; class Derived: public Base { public: void show() { cout << "In Derived \n"; } }; int main(void) { Derived d; Base &br = d; br.show(); return 0; }</code>	Đáp án đúng: In Derived
Câu 147: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream> using namespace std; class P { public: void print() { cout << " Inside P"; } }; class Q : public P { public: void print() { cout << " Inside Q"; } }; class R: public Q { }; int main(void) { R r; r.print(); return 0; }</code>	Đáp án đúng: Inside Q
Câu 148: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: <code>#include<iostream></code> <code>#include<string> using namespace std; class Person { public: void printInfo() { cout << getInfo() << endl; } virtual string getInfo() { return "Person"; } }; class Student: public Person { public: virtual string getInfo() { return "Student"; } }; int main() { Person().printInfo(); Student().printInfo(); return 0; }</code>	Đáp án đúng: Person Student

Câu 149: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include<iostream> #include<string> using namespace std; class A { public: string toString() { return "A"; } }; class B: public A { public: string toString() { return "B"; } }; int main() { A* b = new B(); cout << static_cast<A*>(b)->toString() << b->toString() << endl; return 0; }	Đáp án đúng: AA
Câu 150: Cho biết kết quả xuất hiện trên màn hình sau khi thực thi đoạn chương trình sau: #include <iostream> using namespace std; class A { public: virtual void fun() { cout << "A::fun() "; } }; class B: public A { public: void fun() { cout << "B::fun() "; } }; class C: public B { public: void fun() { cout << "C::fun() "; } }; int main() { B *bp = new C; bp->fun(); return 0; }	Đáp án đúng: C::fun()