



## Kỹ thuật Lập trình

---

### Cuối kì Đề 2

---

nhóm thảo luận CSE  
<https://www.facebook.com/groups/211867931379013>

Tp. Hồ Chí Minh, Tháng 6/2023



## Mục lục

1	Thành phần cơ bản c++	3
2	Hàm và con trỏ	6
3	Danh sách liên kết	12
4	OOP	13
5	Harmony BTL1	19
6	Harmony BT2	19
7	Harmony Thực hành	19



## CUỐI KÌ Đề 2

nhóm thảo luận CSE

<https://www.facebook.com/groups/211867931379013>

### 1 Thành phần cơ bản c++

1. kết quả của đoạn code sau.

```
int main(){  
    cout << "*" << setw(6) << setprecision(4) << 1.23456789 <  
}
```

- A. \* 1.235\*
- B. \*1.234\*
- C. \* 1.234\*
- D. \* 1.23456789\*
- E. Không làm được :))

2. Định danh (tên, biến, hàm...)

- A. được bắt đầu bằng kí tự hoa và thường
- B. Được phép bắt đầu các kí tự \_, !, @, , ..
- C. được bắt đầu bằng số
- D. tất cả đều sai
- E. Không làm được :))

3. kết quả của đoạn code sau.

```
int main(){  
    float f = 0.5 * 16 / 5 - 12 / 7 * 0.2 + 3 / 2.0;  
    cout << f;  
}
```

- A. 2.9
- B. 1.9
- C. 2
- D. 3
- E. Không làm được :))



4. phát biểu nào sau đây là đúng,

```
4  int main(){
5      int* a, b, c[1], *d[1];
6  }
```

- A. a là con trỏ , y là số nguyên, c là mảng số nguyên, d là mảng con trỏ
- B. a là con trỏ , y là con tr, c là mảng số nguyên, d là mảng con trỏ
- C. a là con trỏ , y là con tr, c là mảng con trỏ, d là mảng con trỏ
- D. a là số nguyên , y là con tr, c là mảng con trỏ, d là mảng con trỏ
- E. Không làm được :))

TỪ câu 5 đến câu 7

```
4  struct Pixel{
5      int C, R;
6  };
7
8  void Display(Pixel P){
9      cout << "Col "<< P.C << " Row " << P.R << endl;
10 }
11
12 int main()
13 {
14     Pixel X = {40,50}, Y, Z;
15     Z = X;
16     X.C += 10;
17     Y = Z;
18     Y.C += 10;
19     Y.R += 20;
20     Z.C -= 15;
21     return 0;
22 }
23
```

5. kết quả của *Display(X)*,

- A. Col 50 Row 50
- B. Col 40 Row 50
- C. Col 50 Row 40



- D. Col 40 Row 40  
E. Không làm được :))
6. kết quả của *Display(Y)*,  
A. Col 50 Row 70  
B. Col 40 Row 70  
C. Col 50 Row 50  
D. Col 70 Row 70  
E. Không làm được :))
7. kết quả của *Display(Z)*,  
A. Col 25 Row 50  
B. Col 40 Row 50  
C. Col 45 Row 50  
D. Col 35 Row 50  
E. Không làm được :))
8. kết quả của đoạn code sau.

```
5  ✓ struct sec{
6      int a;
7      char b;
8  };
9  ✓ int main() {
10     struct sec s = {25, 50};
11     struct sec *ps = (struct sec *) &s;
12
13     cout << ps-> a << " " << ps-> b ;
14     return 0;
15 }
16
```

- A. 25 2  
B. 25 50  
C. 25 0  
D. 0 0  
E. Không làm được :))
9. kết quả của đoạn code sau.

```
4
5  int main() {
6      cout << "hello\0hello";
7      return 0;
8  }
9
```

- A. hello



- B. hellohello
- C. hellohe
- D. helloh
- E. Không làm được :))

10. kết quả của đoạn code sau.

```
5  int main() {  
6      short i = 0;  
7      cout << sizeof(i);  
8      return 0;  
9  }  
10
```

- A. 2
- B. 1
- C. 4
- D. 8
- E. Không làm được :))

## 2 Hàm và con trỏ

11. kết quả của đoạn code sau.

```
5  void fun(int arr[], int n){  
6      for(int i = 0; i < n; i++) arr[i] += 1, i += 2;  
7  }  
8  
9  int main() {  
10     int arr[10] = {0,1,2,3,4,5,6,7,8,9};  
11     fun(arr, 10);  
12     for(int i = 0; i < 10; i++) cout << arr[i] << " ";  
13     return 0;  
14 }  
15
```

- A. 1 1 2 4 4 5 7 7 8 10
- B. 1 2 3 4 4 5 7 7 8 10
- C. 0 1 2 4 4 5 7 7 8 10
- D. 1 1 2 4 4 5 7 7 8 9
- E. Không làm được :))



12. kết quả của đoạn code sau.

```
4
5 void fun(int arr[], int n){
6     for(int i = 0; i < n; i++) arr[i] *= 2, i++;
7 }
8
9 int main() {
10     int arr[10] = {0,1,2,3,4,5,6,7,8,9};
11     fun(arr, 10);
12     for(int i = 0; i < 10; i++) cout << arr[i] << " ";
13     return 0;
14 }
15
```

- A. 0 2 4 6 8 10 12 14 16 18
- B. 0 1 2 3 4 5 6 7 8 9
- C. 1 2 4 6 8 10 12 14 16 18
- D. 0 2 2 3 4 5 6 7 8 9
- E. Không làm được :))

13. kết quả của đoạn code sau.

```
4
5 void fun(int *a, int *b){
6     int w;
7     *a = *a + *a;
8     w = *a;
9     *a = *b;
10    *b = w;
11 }
12
13 int main() {
14     int x = 2.0, y = 5.0;
15     int *px = &x, *py = &y;
16     fun(px, py);
17     cout << x << " " << y << endl;
18     return 0;
19 }
20
```

- A. 5 4
- B. 3 4
- C. 1 2
- D. 4 5
- E. Không làm được :))



14. kết quả của đoạn code sau.

```
4  int fun(int = 0, int = 0);
5  int main() {
6      cout << fun(5);
7      return 0;
8  }
9  int fun(int x, int y){
10     return (x + y);
11 }
```

- A. 5
- B. 10
- C. 0
- D. -5
- E. Không làm được :))

15. kết quả của đoạn code sau.

```
4  void lfc(int p)
5  {
6      cout << 1;
7  }
8  void lfc(double q)
9  {
10     cout << 2;
11 }
12 int main(void)
13 {
14     lfc(5);
15     lfc(555.263);
16     return 0;
17 }
```

- A. 12
- B. 21
- C. lỗi biên dịch
- D. lỗi thực thi
- E. Không làm được :))





TỪ câu 16 đến câu 20

```
3
4  int bar(int x, int y){
5      if(y == 0) return 0;
6      return x + bar(x, y - 1);
7  }
8  int foo(int a, int b){
9      if(b == 0) return 1;
10     return bar(a, foo(a, b - 1));
11 }
12
```

16. kết quả của `bar(5,6)`.
- A. 30
  - B. 56
  - C. 65
  - D. lặp vô tận
  - E. Không làm được :))
17. kết quả của `bar(7.5,6.4)`.
- A. 42
  - B. 48
  - C. 56
  - D. lặp vô tận
  - E. Không làm được :))
18. kết quả của `foo(7,1)`.
- A. 7
  - B. 8
  - C. 9
  - D. lặp vô tận
  - E. Không làm được :))
19. kết quả của `foo(2,10)`.
- A. 1024
  - B. 512
  - C. 256
  - D. lặp vô tận
  - E. Không làm được :))
20. kết quả của `foo(2,-1)`.
- A. lặp vô tận
  - B. 0
  - C. 0.5
  - D. 1
  - E. Không làm được :))



21. kết quả của đoạn code sau.

```
int foo(int* s, int* e){
    return *(s + (e - s) / 2);
}

int main()
{
    int array[10] = {1,2,3,4,5,6,7,8,9,10};
    cout << foo(array, array + 9);

    return 0;
}
```

- A. 5
- B. 10
- C. lỗi biên dịch
- D. lỗi thực thi
- E. Không làm được :))

22. kết quả của đoạn code sau.

```
int foo(int* s, int* e){
    return *s ++ + *--e;
}

int main()
{
    int array[10] = {1,2,3,4,5,6,7,8,9,10};
    cout << foo(array, array + 9);

    return 0;
}
```

- A. 10
- B. 11
- C. lỗi biên dịch
- D. lỗi thực thi
- E. Không làm được :))

23. kết quả của đoạn code sau.



```
5  ✓ int main() {  
6  
7      int arr[] = {4, 5, 6, 7};  
8      int *p = (arr + 1);  
9      cout << *p;  
10  
11      return 0;  
12 }
```

- A. 5
- B. 4
- C. lỗi biên dịch
- D. lỗi thực thi
- E. Không làm được :))

24. kết quả của đoạn code sau.

```
4  int main() {  
5  
6      int a[2][4] = {3, 6, 9, 12, 15, 18, 21, 24};  
7      cout << *(a[1] + 2) << ' ' << * ((a + 1) + 3) << ' ' << *(a + 1)[1];  
8      return 0;  
9 }
```

- A. 21 24 1
- B. 15 18 1
- C. lỗi biên dịch
- D. lỗi thực thi
- E. Không làm được :))

25. kết quả của đoạn code sau .

```
4  ✓ int main() {  
5      int a = 8;  
6      a = a >> 1;  
7      a = a << 2;  
8      a |= 15;  
9      a &= 8;  
10     cout << a;  
11 }
```

- A. 8
- B. 16
- C. 32
- D. 2
- E. Không làm được :))



### 3 Danh sách liên kết

Từ câu 26 đến câu 29

```
4 struct Node
5 {
6     int data;
7     Node* next;
8     Node(int data, Node* next){this->data = data, this->next = next;}
9 };
10
11 bool foo(Node* p){
12     if(p == nullptr || p->next == nullptr) return true;
13     return (p->data <= p->next->data) && foo(p->next);
14 }
15
```

26. kết quả khi danh sách liên kết 1->2->3->4->nullptr.

- A. true
- B. false
- C. lỗi
- D. lặp vô tận
- E. Không làm được :))

27. kết quả khi danh sách liên kết 3->2->3->4->nullptr.

- A. false
- B. true
- C. lỗi
- D. lặp vô tận
- E. Không làm được :))

28. kết quả khi danh sách liên kết 3->nullptr.

- A. true
- B. false
- C. lỗi
- D. lặp vô tận
- E. Không làm được :))

29. chức năng của hàm *foo*

- A. danh sách theo thứ tự không giảm
- B. danh sách theo thứ tự không tăng
- C. danh sách theo thứ tự giảm dần
- D. danh sách theo thứ tự tăng dần
- E. Không làm được :))



30. bổ sung đoạn code thiếu của hàm đảo danh sách liên kết .

```
Node* revers(Node* head) {  
    Node* prev = NULL;  
    if(head == NULL) return head;  
    if(head -> next == NULL) return head;  
    Node* curr = head -> next;  
    while(head != NULL) {  
        head -> next = prev;  
        prev = head;  
        head = curr;  
        if(curr != NULL) curr = curr -> next;  
    }  
    //! TODO : your code  
}
```

- A. return prev;
- B. return head;
- C. return curr;
- D. return head->next;
- E. Không làm được :))

## 4 OOP

31. Đây là lý do tốt nhất cho việc sử dụng ngôn ngữ lập trình hướng đối tượng :

- A. Có thể định nghĩa các kiểu dữ liệu riêng và dễ dàng định hình một chương trình hướng đối tượng
- B. Có thể định nghĩa các kiểu dữ liệu riêng
- C. Một chương trình hướng đối tượng có thể được dạy để tự sửa lỗi
- D. Dễ dàng định hình một chương trình hướng đối tượng
- E. Không làm được :))

32. Ba tính năng trong 4 tính năng chính của ngôn ngữ lập trình hướng đối tượng ?

- A. Đóng gói dữ liệu (data encapsulation), thừa kế (inheritance) và đa hình (polymorphism)
- B. Quá tải (overloading), thừa kế (inheritance) và đa hình (polymorphism)
- C. Thừa kế (inheritance), đa hình (polymorphism) và xử lý ngoại lệ (exception handling)
- D. Đóng gói dữ liệu (data encapsulation), thừa kế (inheritance) và xử lý ngoại lệ (exception handling)
- E. Không làm được :))

33. Hàm tạo (constructor) là

- A. một hàm được gọi khi một thực thể của lớp được khởi tạo.
- B. một hàm được gọi khi một thực thể của lớp bị xóa.
- C. một hàm đặc biệt để thay đổi giá trị của bộ nhớ được cấp phát động.
- D. một hàm được gọi để thay đổi giá trị của một biến.
- E. Không làm được :))



34. Hàm tạo (desstructor) là
- A. một hàm được gọi khi một thực thể của lớp bị xóa.
  - B. một hàm được gọi khi một thực thể của lớp được khởi tạo.
  - C. một hàm đặc biệt để thay đổi giá trị của bộ nhớ được cấp phát động.
  - D. một hàm được gọi để thay đổi giá trị của một biến.
  - E. Không làm được :))
35. Giả sử rằng lớp X không có quá tải toán tử cho phép gán (overloaded assignment operator). Điều gì xảy ra khi một phép gán  $a=b$  được thực hiện, với  $a$  và  $b$  là hai đối tượng thuộc lớp X?
- A. Hàm tạo bản sao (copy constructor) sẽ được dùng
  - B. Toán tử gán tự động (automatic assignment operator) sẽ được sử dụng
  - C. Xuất hiện lỗi biên dịch (compiler error)
  - D. Xuất hiện lỗi thực thi (run-time error)
  - E. Không làm được :))
36. nào sau đây là Đa thừa kế
- A.  $A, B \rightarrow C$
  - B.  $A \rightarrow B \rightarrow C$
  - C.  $A \rightarrow B; A \rightarrow C$
  - D.  $B \rightarrow A$
37. để truy xuất thành phần  $x$  bằng con trỏ this
- A.  $this \rightarrow x$
  - B.  $this\_x$
  - C.  $this < -x$
  - D.  $this > x$
38. để truy xuất thành phần  $x$  bằng con trỏ this
- A.  $(*this).x$
  - B.  $*this.x$
  - C.  $this * x;$
  - D.  $*this * x$
39. điền vào đoạn code trống tăng thuộc tính  $x$  của class A lên 1 đơn vị.

```
4 class A{
5     int x;
6 public:
7     void foo(int x){/* TODO your code*/}
8 };
9
```

- A.  $this \rightarrow x += 1;$
- B.  $x += 1$
- C. câu A, B đều đúng
- D. tất cả đều sai
- E. Không làm được :))



Từ câu 40 đến câu 41

```
4  class A{
5      int d;
6      void foo(){}
7  public:
8      int a, b;
9  protected:
10     int c;
11 };
12
13 class B : public A{}
14
```

40. cách truy cập hàm *foo* của class A trong class B.

- A. tất cả đều sai
- B. A::foo();
- C. this->foo();
- D. foo();
- E. Không làm được :))

41. class B có thể truy cập các thuộc tính nào của class A

- A. a, b, c
- B. a, b
- C. a, b, c, d
- D. a, b, d
- E. Không làm được :))

42. đối tượng của a trong hàm foo của class B có thể truy cập các thuộc tính nào của class A. chọn câu đúng nhất.

```
4  class A{
5  private:
6      int a;
7  protected:
8      int b;
9  public:
10     int c;
11     friend class B;
12 };
13
14 class B{
15 public:
16     foo(A a){}
17 }
```



- A. a, b, c;
- B. không truy cập được a
- C. không truy cập được b
- D. không truy cập được a, b
- E. Không làm được :))

43. chọn câu đúng nhất.

- A. con trở lớp dẫn xuất không thể trở đến lớp cơ sở
- B. không thể khai báo con trở dẫn xuất
- C. con trở lớp cơ sở không thể trở đến lớp dẫn xuất
- D. không thể khai báo con trở lớp cơ sở
- E. Không làm được :))

Từ câu 44 đến câu 451

```
4  class A{
5  public:
6      A(){}
7      ~A(){}
8      void f1(){}
9      virtual void f2() = 0;
10 };
11
```

44. class B thừa kế từ class A thì class B bắt buộc phải hiện thực hàm nào

- A. f2
- B. f1
- C. constructor
- D. destructor
- E. Không làm được :))

45. class B như câu trên thì khai báo nào sau đây là sai

- A. A a;
- B. B b;
- C. A\* a = new B();
- D. B\* a = new B();
- E. Không làm được :))

46. kết quả đoạn code sau.





```
4  class A{
5      public: A() {cout<<"A";}
6      ~A(){cout << "a";}
7  };
8  class B{
9      A a;
10     public:
11     B() {cout<<"B";}
12     ~B(){cout << "b";}
13 };
14
15 int main() {
16     B b;
17 }
```

- A. ABba
- B. AaBb
- C. ABab
- D. AB
- E. Không làm được :))

47. dòng code bị lỗi.

```
4  class watch {
5      void watch() {
6          hour = 0;
7          minute = 1;
8      }
9      private:
10     int hour = 12;
11     int minute;
12 };
13
14
```

- A. 5
- B. 6
- C. 7
- D. 9
- E. Không làm được :))

48. dòng code bị lỗi.



```
4  class watch {
5  private:
6      int hour = 12;
7      int minute;
8  public:
9      watch(int hour, int minute)
10     : hour(0){
11         this.minute = minute;
12     }
13 };
14
```

- A. 11
- B. 10
- C. 6
- D. 7
- E. Không làm được :))

49. kết quả đoạn code sau.

```
class A{
    public: A() {cout<<"A";}
    ~A(){cout << "a";}
};

class B{
    A a;
public:
    B() {cout<<"B";}
    ~B(){cout << "b";}
};

class C: B{
    public: C() {cout<<"C";}
    ~C(){cout << "c";}
};

int main() {
    C* c = new C();
    delete c;
}
```

- A. ABCcba
- B. AaCcBb



- C. ABCcab
- D. ABC
- E. Không làm được :))

50. nào sau đây không phải access modifier.

- A. friend
- B. public
- C. private
- D. protected
- E. Không làm được :))

## 5 Harmony BTL1

- 1) tìm vị trí số lớn thứ 2 trong 3 số đầu tiên trong mảng hiện thực hàm `int BTL1(int arr[], int n)` hàm trả về vị trí của số lớn thứ 2. VD mảng 1 2 3 4 5 6 vị trí số lớn thứ 2 trong 3 số đầu là `index = 1`
- 2) luôn tăng số `n` lên số nguyên tố gần nhất và tiếp tục luôn giảm `n` để số fibnacion gần nhất, hàm có sẵn `bool isSoNguyenTo(int n)`; hãy hiện thực hàm `int BTL1(int n)` trả về số `n` đã thay đổi

## 6 Harmony BT2

- 1) hiện thực lại sự kiện 99
- 2) hiện thực lại cách truyền vàng của hiệp sĩ

## 7 Harmony Thực hành

này chưa có đề để sau

nhóm thảo luận CSE

<https://www.facebook.com/groups/211867931379013>

CHÚC CÁC EM THI TỐT

