



## Kỹ thuật Lập trình

---

### Cuối kì Đề 1

---

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	Hướng đối tượng	12
4	Harmony BTL1	21
5	Harmony BT2	21
6	Harmony Thực hành	22



# CUỐI KÌ ĐỀ 1

## nhóm thảo luận CSE

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

### 1 Thành phần cơ bản c++

Điều kiện câu 1 đến 5 `int arr[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8}`,  $i = 3$

1. Giá trị của phần tử có `arr[9]`.
  - A. 9
  - B. 0
  - C. random
  - D. lỗi
2. câu lệnh nào sau đây không hợp lệ.
  - A. `arr[-i]`
  - B. `arr[arr[i]]`
  - C. `arr[i - 4]`
  - D. `arr[i++]`
3. không phải địa chỉ của `arr[3]`.
  - A. `&arr[0] + 3`
  - B. `arr + 3`
  - C. `++arr + 2`
  - D. `++arr + 3`
4. kết quả của đoạn code sau.

```
int main(){
    int arr[10] = {0,1,2,3,4,5,6,7,8};
    int x = 0;
    for(int i = 2; i < 9; i++){
        x += arr[i];
        arr[i] += arr[i - 1] + arr[i - 2];
    }
    cout << x;
}
```

- A. 35
- B. 34
- C. 30
- D. 40



5. kết quả của đoạn code sau.

```
int main(){
    int arr[10] = {0,1,2,3,4,5,6,7,8};
    int* start = &arr[0];
    int* end = &arr[0] + 9;
    cout << *((end - start) / 2 + start);
}
```

- A. 3
- B. 4
- C. 5
- D. 6

6. khai báo chuỗi nào sau đây là đúng cho "abcd".

- A. char arr[] = {"a", "b", "c", "d"};
- B. char arr[] = {'a', 'b', 'c', 'd'};
- C. char arr[] = "abcd";
- D. char arr[] = 'abcd';

7. khai báo mảng 2 chiều nào đúng

- A. int arr[3][] = {{1,2,3}, {4,5,6}, {7,8,9}};
- B. int arr[][2] = {{1,2,3}, {4,5,6}, {7,8,9}};
- C. int arr[][3] = {{1,2,3}, {4,5,6}, {7,8,9}};
- D. int arr[2][3] = {{1,2,3}, {4,5,6}, {7,8,9}};

8. kết quả của đoạn code sau.

```
7 int main(){
8     cout << sizeof(int) << " "
9         << sizeof(bool) << " "
10        << sizeof(float) << " "
11        << sizeof(long long) << " "
12        << sizeof(double);
13 }
14
```

- A. 4 1 4 8 8
- B. 4 1 8 4 8
- C. 4 1 8 4 4
- D. 4 1 4 8 4



9. kết quả của đoạn code sau.

```
6
7  int main(){
8      enum season
9      { spring = 0,
10         summer = 4,
11         autumn = 8,
12         winter
13     };
14
15     season a = summer;
16     season b = winter;
17
18     cout << a << " " << b << endl;
19
20 }
21
```

- A. 4 9
- B. 4 12
- C. 2 4
- D. 2 12

10. kết quả của đoạn code sau.

```
7  int main(){
8      enum season
9      { spring = 0,
10         summer,
11         autumn,
12         winter,
13     };
14
15     season a = summer;
16     season b = winter;
17
18     cout << a << " " << b << endl;
19
20 }
21
```

- A. 0 5
- B. 1 3



- C. 1 2
- D. 0 3

## 2 Hàm và con trỏ

11. Hàm nào sau đây hợp lệ.
- A. `void func(void){return 0.0;}`
  - B. `int func(int a = 0, int b, float c) {return 0;}`
  - C. `int func(int a = 0, int b = 0, float c = 0.0) {return 0;}`
  - D. `void func(int a*b, float c) {return 0;}`
12. Giả sử ta có lời gọi hàm sau: `fun(a + b, 3, max(n - 1, b, a*b+2))`;  
Số lượng tham số thực của hàm `fun` là.
- A. 3
  - B. 4
  - C. 5
  - D. 6
- từ câu 13 đến 16 : cho các hàm sau
- (1) `func(int a, float b, int c=0);`
  - (2) `func(float a, int b, int c=0);`
  - (3) `func(float a, float b);`
13. Giả sử ta có lời gọi hàm sau: `func(1,1,0,5)`;  
hàm nào sẽ được gọi
- A. 1
  - B. 2
  - C. 3
  - D. lỗi
14. Giả sử ta có lời gọi hàm sau: `func(1.0,1)`;  
hàm nào sẽ được gọi
- A. 1
  - B. 2
  - C. 3
  - D. lỗi
15. Giả sử ta có lời gọi hàm sau: `func(1,1,5)`;  
hàm nào sẽ được gọi
- A. 1
  - B. 2
  - C. 3
  - D. lỗi
16. Giả sử ta có lời gọi hàm sau: `func(1.0,1.0)`;  
hàm nào sẽ được gọi
- A. 1
  - B. 2
  - C. 3
  - D. lỗi



17. Giả sử ta khai báo “int \*p, str[10];”, cho biết câu lệnh nào dưới đây đúng cú pháp

- A. p = &str;
- B. p = str;
- C. str++;
- D. str = p;

18. kết quả của đoạn code sau.

```
5 void func(int a, int b){
6     a ++;
7     b ++;
8 }
9
10 int main(){
11     int a = 2, b = 3;
12     func(a, b);
13     cout << a << b << endl;
14 }
15
```

- A. 2 3
- B. 3 4
- C. 34
- D. 23

19. kết quả của đoạn code sau.

```
5 void func(int a, int& b){
6     a ++;
7     b ++;
8 }
9
10 int main(){
11     int a = 2, b = 3;
12     func(a, b);
13     cout << a << " " << b << endl;
14 }
15
```

- A. 2 3
- B. 2 4
- C. 3 4
- D. 23



20. chức năng của hàm này.

```
5 bool func(int a, int b){  
6     return a > b ? 1 : 0;  
7 }  
8
```

- A. kiểm tra  $a \geq b$  hay không.
- B. kiểm tra  $a \leq b$  hay không.
- C. kiểm tra  $a > b$  hay không.
- D. kiểm tra  $a < b$  hay không.

21. cho  $a = 2$  thì  $a \ll 2$  bằng nhiêu

- A. 2
- B. 8
- C. 16
- D. 0

22. kết quả của đoạn code sau.

```
9 int main(){  
10     int a = 2, b = 1;  
11  
12     if(a > b && a ++ || ++b){  
13         a *= 2;  
14     }  
15     cout << a << " " << b << endl;  
16 }  
17
```

- A. 2 1
- B. 6 2
- C. 6 1
- D. 3 2

23. kết quả của đoạn code sau.

```
int main(){  
    int a = 2, b = 1;  
  
    if(a > b || a ++ && ++b){  
        a *= 2;  
    }  
    cout << a << " " << b << endl;  
}
```





- A. 2 1
- B. 6 2
- C. 4 1
- D. 3 2

24. kết quả của `func(4)`.

```
5  ✓ int func(int a){  
6      if(a == 0) return 1;  
7      return a * a + func(a - 1);  
8  }  
9
```

- A. 21
- B. 25
- C. 31
- D. 30

25. kết quả của `func(5)`.

```
5  ✓ int func(int a){  
6      if(a == 0) return 1;  
7      return a / 2 + func(a - 1);  
8  }  
9
```

- A. 3
- B. 4
- C. 5
- D. 6

26. kết quả của `func(5)`.

```
4  
5  int fund(int a){  
6      return (a == 0) ? 0 : 1 + fund(a - 1);  
7  }  
8  
9  int func(int a){  
10     if(a == 1) return 1;  
11     return fund(a) * func(a - 1);  
12 }  
13
```

- A. 720
- B. 21
- C. 120



D. 20

27. kết quả .

```
int main(){  
    int a = 2.0 / 4 + 6 / 4 + 1.2 * 3 + 6 / 5 * 1.0;  
    cout << a << endl;  
}
```

- A. 6
- B. 5
- C. 5.1
- D. 6.2

28. kết quả.

```
int main(){  
    int a = 1;  
    int b = a ? 2 : 3;  
    cout << a << " " << b << endl;  
}
```

- A. 1 2
- B. 1 3
- C. 0 2
- D. 0 3

29. kết quả.

```
13  
14 int main(){  
15     int a = 1;  
16     int b = (a = 0) ? 2 : 3;  
17     cout << a << " " << b << endl;  
18 }  
19
```

- A. 1 2
- B. 1 3
- C. 0 2
- D. 0 3

30. kết quả.



```
14 int main(){
15     cout << 15 % 4 << " "
16         << 15 % -4 << " "
17         << (- 15 % 4) << " "
18         << - 15 % -4 << endl;
19 }
20
```

- A. 3 3 -3 -3
- B. 3 3 3 3
- C. 3 3 -3 3
- D. 3 3 3 -3

31. kết quả.

```
14 int main(){
15     int a[][4]={1,2,3,4,5,6,7,8,9,10,11};
16     for (int i = 0; i < 4; i++)
17         cout << a[2][i] << " ";
18 }
19
```

- A. 1 2 3 4
- B. 5 6 7 8
- C. 9 10 11 0
- D. 9 10 11 12



### 3 Hướng đối tượng

Cho đoạn code sau từ câu 32 đến 41.

```
5  class Base{
6  private:
7      int a = 0;
8  protected:
9      int b = 0;
10 public:
11     int c = 0;
12 public:
13     Base(int a, int b, int c): b(b){
14         this->a = a;
15         c = c;
16     }
17
18     void print(){
19         cout << a << " " << b << " " << c;
20     }
21 };
22
```

32. `Base* a = new Base(1,2,3)` kết quả của `a.print()`

- A. 1 2 3
- B. 1 2 0
- C. 1 0 3
- D. 0 2 3

33. đoạn code sau sai gì.

```
24  int main(){
25      Base* a = new Base(1,2,3);
26      cout << a->c;
27      cout << a->a;
28      cout << a->b;
29  }
30
```

- A. a b nằm ngoài phạm vi
- B. a c nằm ngoài phạm vi
- C. b b nằm ngoài phạm vi
- D. không lỗi



34. đoạn code sau sai gì.

```
23  
24 int main(){  
25     Base* a = new Base(1, 2, 3);  
26     Base* b = new Base(a);  
27     Base* c = a;  
28 }  
29
```

- A. lỗi dòng 25
- B. lỗi dòng 26
- C. lỗi dòng 27
- D. không lỗi

35. đoạn code sau sai gì.

```
22  
23 int main(){  
24     Base a = Base(1, 2, 3);  
25     Base b(1, 2, 3);  
26     Base d(a);  
27     Base c = b;  
28  
29
```

- A. lỗi dòng 25
- B. lỗi dòng 26
- C. lỗi dòng 27
- D. không lỗi

Cho đoạn code sau 2 câu tiếp theo.



```
22
23  class Derived : public Base{
24  public:
25      void printf(){
26          cout << a;
27          cout << b;
28          cout << c;
29      }
30  };
31
32  class PDerived : public Derived{
33  public:
34      void printf(){
35          cout << a;
36          cout << b;
37          cout << c;
38      }
39  };
40
```

36. câu nào đúng nhất class Derived.

- A. a b không thể truy cập
- B. a không thể truy cập
- C. b không thể truy cập
- D. c không thể truy cập

37. câu nào đúng nhất class PDerived.

- A. a b không thể truy cập
- B. a không thể truy cập
- C. b không thể truy cập
- D. c không thể truy cập

Cho đoạn code sau 2 câu tiếp theo.



```
23 class Derived : protected Base{
24 public:
25     void printf(){
26         cout << a;
27         cout << b;
28         cout << c;
29     }
30 };
31
32 class PDerived : protected Derived{
33 public:
34     void printf(){
35         cout << a;
36         cout << b;
37         cout << c;
38     }
39 };
40
```

38. câu nào đúng nhất class Derived.

- A. a b không thể truy cập
- B. a không thể truy cập
- C. b không thể truy cập
- D. c không thể truy cập

39. câu nào đúng nhất class PDerived.

- A. a b không thể truy cập
- B. a không thể truy cập
- C. b không thể truy cập
- D. c không thể truy cập

Cho đoạn code sau 2 câu tiếp theo.



```
22
23 class Derived : Base{
24 public:
25     void printf(){
26         cout << a;
27         cout << b;
28         cout << c;
29     }
30 };
31
32 class PDerived : Derived{
33 public:
34     void printf(){
35         cout << a;
36         cout << b;
37         cout << c;
38     }
39 };
40
```

40. câu nào đúng nhất class Derived.

- A. a b không thể truy cập
- B. a không thể truy cập
- C. b không thể truy cập
- D. c không thể truy cập

41. câu nào đúng nhất class PDerived.

- A. a b c không thể truy cập
- B. a không thể truy cập
- C. b không thể truy cập
- D. c không thể truy cập

Cho đoạn code sau từ câu 42 đến 40.





```
5  class A{
6  public:
7      A(){cout << " Constructor class A" << endl;}
8      virtual void print1(){cout << "A1" << endl;}
9      void print2(){cout << "A2" << endl;}
10     ~A(){cout << " Destructor class A" << endl;}
11
12 };
13
14 class B: public A{
15 public:
16     B(){cout << " Constructor class B" << endl;}
17     virtual void print1(){cout << "B1" << endl;}
18     void print2(){cout << "B2" << endl;}
19     ~B(){cout << " Destructor class B" << endl;}
20 };
21
22 class C: public B{
23 public:
24     C(){cout << " Constructor class C" << endl;}
25     virtual void print1(){cout << "C1" << endl;}
26     void print2(){cout << "C2" << endl;}
27     ~C(){cout << " Destructor class C" << endl;}
28 };
29
```

42. khai báo `C* x = new C` thứ tự gọi hàm contrutor;

- A. class a -> class b -> class c
- B. class c -> class b -> class a
- C. class a -> class c -> class b
- D. class c -> class a -> class b

43. khai báo `delete x` thứ tự gọi hàm destrutor;

- A. class a -> class b -> class c
- B. class c -> class b -> class a
- C. class a -> class c -> class b
- D. class c -> class a -> class b

44. kết quả `x->print1()` và `x->print2()`;

- A. C1 và C2
- B. A1 và A2
- C. B1 và B2
- D. C1 và A2

45. khai báo `A* y = new C` thứ tự gọi hàm contrutor;

- A. class a -> class b -> class c
- B. class c -> class b -> class a
- C. class a -> class c -> class b



- D. class c -> class a -> class b
46. khai báo *delete y* thứ tự gọi hàm destrutor;
- A. class a -> class b -> class c
  - B. class c
  - C. class a
  - D. class b
47. kết quả *y- > print1()* và *y- > print2()*;
- A. C1 và C2
  - B. A1 và A2
  - C. B1 và B2
  - D. C1 và A2
48. khai báo *delete y* ta có thể thấy không gọi hết các hàm hủy hãy đề xuất giải pháp thiết kế
- A. thêm virtual cho class hủy A
  - B. thêm virtual cho class hủy A, B
  - C. thêm virtual cho class hủy B
  - D. thêm virtual cho class hủy C
49. nếu bỏ đoạn code 17 virtual void print1(){cout << "B1" << endl;} thì kết quả của câu 47
- A. C1 và C2
  - B. A1 và A2
  - C. lỗi
  - D. C1 và A2
50. nếu bỏ đoạn code 26 void print2(){cout << "C2" << endl;} thì kết quả của câu 44
- A. C1 và C2
  - B. A1 và A2
  - C. C1 và B2
  - D. C1 và A2
51. sửa đoạn code 8 từ virtual void print1()cout << "A1" << endl; sang virtual void print1() = 0; thì kết quả của câu 47
- A. C1 và C2
  - B. A1 và A2
  - C. lỗi
  - D. C1 và A2
52. giả sử thiết kế thêm 1 class D được thừa kế từ class A thì class D cần có hàm gì để trường trình chạy được ( chưa nói đến tính đúng đắn của OOP)
- A. Contructor
  - B. Destructor
  - C. print1()
  - D. print2()



53. đoạn code sau sai gì.

```
31 int main(){
32     A* d = new C;
33     A* e = new B;
34     A* f = new A;
35
36     B* m = new C;
37     B* n = new B;
38     B* k = new A;
39
40     C* x = new C;
41     C* y = new B;
42     C* z = new A;
43 }
44
```

- A. lỗi dòng 38 41 42
- B. lỗi dòng 37 38 41 42
- C. lỗi dòng 37 38 40 41 42
- D. không lỗi

54. đoạn code sau sai gì.

```
5 class MyStaticClass{
6 public:
7     static int value;
8     MyStaticClass(){ value ++;}
9 };
10 int MyStaticClass::value = 0;
11 int main(){
12     for(int i = 0; i < 10; i++ )
13         MyStaticClass *c = new MyStaticClass();
14     cout << MyStaticClass::value << endl;
15 }
16
```

- A. 10
- B. 0
- C. 1
- D. lỗi

55. đoạn code sau sai gì.



```
5  class MyStaticClass{
6  public:
7      static int value;
8      MyStaticClass(){ value ++;}
9  };
10 int MyStaticClass::value = 0;
11 int main(){
12     for(int i = 0; i < 10; i++ ){
13         MyStaticClass *c = new MyStaticClass();
14         cout << c->value << " ";
15     }
16 }
17 ~
```

- A. 1 2 3 4 5 6 7 8 9 10
- B. 0 0 0 0 0 0 0 0 0 0
- C. 0 1 2 3 4 5 6 7 8 9
- D. lỗi

56. đoạn code sau sai gì.

```
#include <bits/stdc++.h>
using namespace std;
class A {
private:
    int a = 10;
public:
    friend void print(A x);
};
void print(A x)
{
    cout << x.a;
}

int main()
{
    A x;
    print(x);
    return 0;
}
```

- A. lỗi biên dịch



- B. lỗi thực thi  
C. 10  
D. 11
57. class A là bạn của class B, class B là bạn của class C thì:  
A. class C có thể truy cập thành viên private của class A  
B. class A có thể truy cập thành viên private của class C  
C. class A có thể truy cập thành viên private của class B  
D. class C có thể truy cập thành viên private của class B
58. class A là bạn của class B, class C là con của class A:  
A. class C có thể truy cập thành viên private của class A  
B. class A có thể truy cập thành viên private của class C  
C. class A có thể truy cập thành viên private của class B  
D. class C có thể truy cập thành viên private của class B
59. class A là bạn của class B, class C là con của class B:  
A. class C có thể truy cập thành viên private của class A  
B. class A có thể truy cập thành viên private của class C  
C. class A có thể truy cập thành viên private của class B  
D. class C có thể truy cập thành viên private của class B
60. kích thước của class A

```
5  class A{  
6      int a;  
7      bool c;  
8  };  
9
```

- A. 8  
B. 7  
C. 5  
D. 4

## 4 Harmony BTL1

- 1) cho mảng `array[]` có n phần tử hãy tăng các phần tử chẵn trong luôn giảm xuống hp xuống fibnacion gần nhất và luôn tăng phần tử lẻ lên số nguyên tố gần nhất, mọi phần tử đều  $> 2$ , hãy hiện thức hàm `void BTL1(int array[], int n);`  
2) cho mảng `array[]` có n phần tử hãy kiểm tra mảng có phải dạng hình núi hay không

## 5 Harmony BT2

- 1) hiện thực lại hàm create của class Baseknight  
2) hiện thức lại get trong class BaseBag



## 6 Harmony Thực hành

này chưa có đề để sau

nhóm thảo luận CSE

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

CHÚC CÁC EM THI TỐT

