

<b>Trạng thái</b>	Đã xong
<b>Bắt đầu vào lúc</b>	Thứ Tư, 7 tháng 5 2025, 2:54 AM
<b>Kết thúc lúc</b>	Thứ Tư, 7 tháng 5 2025, 3:08 AM
<b>Thời gian thực hiện</b>	14 phút 22 giây

## Câu hỏi 1

Đúng

Hãy xem xét câu lệnh:

```
ClockType myClock (5, 12, 40);
```

Câu lệnh này khai báo một đối tượng myClock thuộc class ClockType. Ở đây, chúng ta đang truyền ba giá trị kiểu **int**, giá trị này khớp với kiểu của các tham số chính thức của hàm tạo với một tham số.

- Xem xét đoạn code cho trước trong phần trả lời, chú ý đến hàm khởi tạo có 3 tham số. Hãy hiện thực hàm này để sau khi gọi câu lệnh khai báo trên, 3 biến thành viên được lần lượt đặt thành 5, 12, 40.
- Hiện thực hàm khởi tạo với không tham số, hàm sẽ gán giá trị **0** cho 3 biến thành viên.

**Lưu ý:** hr, min, sec cần thoả mãn các điều kiện sau. Nếu tham số đầu vào không thoả mãn điều kiện bên dưới thì ta gán giá trị 0 cho biến thành viên tương ứng.

- $0 \leq \text{hr} < 24$
- $0 \leq \text{min} < 60$
- $0 \leq \text{sec} < 60$

[English]

Consider the following statement:

```
clockType myClock (5, 12, 40);
```

This statement declares a variable name myClock belongs to type ClockType with 3 input variables.

- Complete the constructor with 3 parameters in the class given in the answer box so that the time can be printed correctly.
- Complete the constructor with 0 parameters that set the value of three attributes to 0.

**Note:** hr, min, sec need to satisfy the following conditions. If the input parameter does not satisfy the condition below, we assign the value 0 to the corresponding member variable:

- $0 \leq \text{hr} < 24$
- $0 \leq \text{min} < 60$
- $0 \leq \text{sec} < 60$

For example:

Test	Result
ClockType myClock; myClock.printTime();	00:00:00

**Answer:** (penalty regime: 0 %)

Reset answer

```

1  class ClockType
2  {
3  public:
4      ClockType(int, int, int); //constructor with parameters
5      ClockType(); //default constructor
6      void printTime() const;
7  private:
8      int hr;
9      int min;
10     int sec;
11 };
12
13 // Hàm khởi tạo với 3 tham số
14 ClockType::ClockType(int hours, int minutes, int seconds)
15 {
16     // Kiểm tra điều kiện hợp lệ cho giờ
17     if (hours >= 0 && hours < 24)
18         hr = hours;
19     else
20         hr = 0;
21
22     // Kiểm tra điều kiện hợp lệ cho phút
23     if (minutes >= 0 && minutes < 60)
24         min = minutes;
25     else

```

```

26         min = 0;
27
28         // Kiểm tra điều kiện hợp lệ cho giây
29         if (seconds >= 0 && seconds < 60)
30             sec = seconds;
31         else
32             sec = 0;
33     }
34
35     // Hàm khởi tạo mặc định không có tham số
36     ClockType::ClockType()
37     {
38         hr = 0;
39         min = 0;
40         sec = 0;
41     }
42
43     void ClockType::printTime() const {
44         if (hr < 10)
45             cout << "0";
46         cout << hr << ":";
47         if (min < 10)
48             cout << "0";
49         cout << min << ":";
50         if (sec < 10)
51             cout << "0";
52         cout << sec;

```

	Test	Expected	Got	
✓	ClockType myClock; myClock.printTime();	00:00:00	00:00:00	✓
✓	ClockType yourClock(5,2,30); yourClock.printTime();	05:02:30	05:02:30	✓

Passed all tests! ✓

## Câu hỏi 2

Đúng

Cho định nghĩa class:

```
class ClockType
{
    public:
        ClockType();
        void setTime(int, int, int);
        void printTime() const;
    private:
        int hr;
        int min;
        int sec;
};
```

Cho câu lệnh:

```
myClock.setTime(5, 2, 30);
```

Trong câu lệnh **myClock.setTime (5, 2, 30)**: phương thức setTime được thực thi. Các giá trị 5, 2 và 30 được chuyển dưới dạng tham số cho hàm setTime và hàm sử dụng các giá trị này để đặt giá trị của ba biến thành viên hr, min và sec của đối tượng myClock thành 5, 2 và 30, tương ứng.

**Yêu cầu:** SV hiện thực phương thức **setTime** để nó thực hiện được mô tả trên.

**Lưu ý:** hr, min, sec cần thoả mãn các điều kiện sau. Nếu tham số đầu vào không thoả mãn điều kiện bên dưới thì ta gán giá trị 0 cho biến thành viên tương ứng.

- $0 \leq \text{hr} < 24$
- $0 \leq \text{min} < 60$
- $0 \leq \text{sec} < 60$

[English]

Given the class definition:

```
class ClockType
{
    public:
        ClockType();
        void setTime(int, int, int);
        void printTime() const;
    private:
        int hr;
        int min;
        int sec;
};
```

And the following command:

```
myClock.setTime(5, 2, 30);
```

In the statement **myClock.setTime(5, 2, 30)**: the method setTime is executed. The values 5, 2, and 30 are passed as parameters to the function setTime, and the function uses these values to set the values of the three member variables hr, min, and sec of object myClock to 5, 2, and 30, respectively.

**Requirement:** Student implements the **setTime** method for it to do as described above.

**Note:** hr, min, sec need to satisfy the following conditions. If the input parameter does not satisfy the condition below, we assign the value 0 to the corresponding member variable:

- $0 \leq \text{hr} < 24$
- $0 \leq \text{min} < 60$
- $0 \leq \text{sec} < 60$

For example:

Test	Result
<pre>ClockType myClock; myClock.setTime(5, 4, 30); myClock.printTime();</pre>	05:04:30

Answer: (penalty regime: 0 %)

Reset answer

```

1  class ClockType
2  {
3  public:
4      ClockType();
5
6      void setTime(int, int, int);
7      void printTime() const;
8
9  private:
10     int hr;
11     int min;
12     int sec;
13 };
14
15 void ClockType::printTime() const
16 {
17     if (hr < 10)
18         cout << "0";
19     cout << hr << ":";
20     if (min < 10)
21         cout << "0";
22     cout << min << ":";
23     if (sec < 10)
24         cout << "0";
25     cout << sec;
26 }
27
28 // TODO
29 void ClockType::setTime(int hours, int minutes, int seconds)
30 {
31     // Kiểm tra điều kiện hợp lệ cho giờ
32     if (hours >= 0 && hours < 24)
33         hr = hours;
34     else
35         hr = 0;
36
37     // Kiểm tra điều kiện hợp lệ cho phút
38     if (minutes >= 0 && minutes < 60)
39         min = minutes;
40     else
41         min = 0;
42
43     // Kiểm tra điều kiện hợp lệ cho giây
44     if (seconds >= 0 && seconds < 60)
45         sec = seconds;
46     else
47         sec = 0;
48 }
49

```

	Test	Expected	Got	
✓	<pre>ClockType myClock; myClock.setTime(5, 4, 30); myClock.printTime();</pre>	05:04:30	05:04:30	✓
✓	<pre>ClockType myClock; myClock.setTime(10, 12, 60); myClock.printTime();</pre>	10:12:00	10:12:00	✓

Passed all tests! ✓

## Câu hỏi 3

Đúng

Cho class template Array như bên dưới dùng để chứa một mảng (1 chiều) được khai báo trong vùng nhớ Heap. Trong class Array có khai báo một số phương thức (hàm) để thao tác với Array.

```
template <typename T>
class Array {
public:
    Array(int size, T initValue);
    ~Array();
    void print();
```

```
    void setAt(int idx, const T & value);
    T getAt(int idx);
    T& operator[](int idx);
```

```
private:
    int size;
    T * p;
};
```

Trong class Array có khai báo các thuộc tính sau:

- Thuộc tính **p** là con trỏ trỏ đến vùng nhớ Heap được cấp phát.
- Thuộc tính **size** của Array chứa số lượng phần tử của mảng.

**Yêu cầu:** SV hiện thực phương thức được mô tả như sau:

- **operator[](int idx):** quá tải toán tử [] để thực hiện được cả 2 việc: đọc (lấy) giá trị của phần tử ở vị trí idx và ghi (gán) một giá trị mới vào phần tử ở vị trí idx. Ví dụ: đọc:  $x = a[2]$ , ghi:  $a[2] = 5$  với  $a$  là một đối tượng của class Array. Nếu idx có giá trị không hợp lệ ( $idx < 0$  hoặc  $idx \geq size$ ) thì **throw -1**;

[English]

Given class template **Array** as below that contains an array allocated in Heap memory.

```
template <typename T>
class Array {
public:
    Array(int size, T initValue);
    ~Array();
    void print();
```

```
    void setAt(int idx, const T & value);
    T getAt(int idx);
    T& operator[](int idx);
```

```
private:
    int size;
    T * p;
};
```

There are 2 attributes in class Array:

- **p**: a pointer contains the address of first element of allocated memory.
- **size**: number of elements of array.

**Requirement:** Implement following method:

- **operator[](int idx):** overload operator [] to do: read (get) the value of element at position idx, and write (set) a new value to the element at position idx. For instance: read:  $x = a[2]$ , write:  $a[2] = 5$ , for  $a$  is an object of class Array. if idx has an invalid value ( $idx < 0$  or  $idx \geq size$ ), then throw -1;

**For example:**

[illegible]

**Answer:** (penalty regime: 0 %)

Reset answer

```

1  template <typename T>
2  class Array {
3  public:
4      Array(int size, T initialValue);
5      ~Array();
6
7      void setAt(int idx, const T & value);
8      T getAt(int idx);
9      T& operator[](int idx);
10
11     void print();
12
13 private:
14     int size;
15     T * p;
16 };
17
18 template<typename T>
19 void Array<T>::print() {
20     for (int i = 0; i < this->size; ++i) {
21         cout << (i > 0 ? " " : "")
22             << this->p[i];
23     }
24     cout << endl;
25 }
26
27 // TODO
28 template <typename T>
29 T& Array<T>::operator[](int idx) {
30     // Kiểm tra chỉ số idx có hợp lệ hay không
31     if (idx < 0 || idx >= size) {
32         throw -1; // Ném ngoại lệ -1 nếu chỉ số không hợp lệ
33     }
34
35     // Trả về tham chiếu đến phần tử tại vị trí idx
36     return p[idx];
37 }
38

```



[illegible]

Passed all tests! ✓

## Câu hỏi 4

Đúng

Hãy thiết kế một class **Integer** với một biến private **val** dạng **int**, Class gồm các phương thức sau:

- **constructor** nhận biến **val** ban đầu.
- **setValue()** để thay đổi giá trị biến **val**.
- **getValue()** để lấy giá trị biến **val**.

Define a class Integer with one private variable **val** type int with:

- \_ A constructor with one input for **val**.
- \_ A function setValue() to change the value of **val**.
- \_ A function getValue() to return the value of **val**.

For example:

Test	Result
<pre>cout&lt;&lt;"Constructor test"; Integer i(10); Integer i2(-10); Integer *i3=new Integer(20); delete i3;</pre>	Constructor test

Answer: (penalty regime: 0 %)

```
1 class Integer {
2     private:
3         int val;
4
5     public:
6         // Constructor nhận biến val ban đầu
7         Integer(int initialValue) {
8             val = initialValue;
9         }
10
11        // Phương thức setValue() để thay đổi giá trị biến val
12        void setValue(int newValue) {
13            val = newValue;
14        }
15
16        // Phương thức getValue() để lấy giá trị biến val
17        int getValue() const {
18            return val;
19        }
20    };
```

	Test	Expected	Got	
✓	cout<<"Constructor test"; Integer i(10); Integer i2(-10); Integer *i3=new Integer(20); delete i3;	Constructor test	Constructor test	✓
✓	//getValue Test Integer i(10); cout << i.getValue();	10	10	✓

Passed all tests! ✓

## Câu hỏi 5

Đúng

Hãy thiết kế một class Room với constructor gồm 3 biến length, breadth, height (theo thứ tự). Hiện thực các phương thức sau cho class Room:

- **Constructor:** đầu vào lần lượt là **length, breadth, height**
- **calculateArea:** tính diện tích của căn phòng.
- **calculateVolume:** tính thể tích của căn phòng

Define class Room with three variable length, breadth and height and:

- \_ A constructor with three input for length, breadth and height.
- \_ Function calculateArea to calculate and return the room area.
- \_ Function calculateVolume to calculate and return the room volume.

For example:

Test	Result
cout<<"Constructor test"; Room r(20,3,4); Room *r2=new Room(10.5,5.5,5.4); delete r2;	Constructor test

Answer: (penalty regime: 0 %)

```
1 class Room {
2     private:
3         double length;
4         double breadth;
5         double height;
6
7     public:
8         // Constructor với 3 tham số length, breadth, height
9         Room(double length, double breadth, double height) {
10             this->length = length;
11             this->breadth = breadth;
12             this->height = height;
13         }
14
15         // Phương thức calculateArea() tính diện tích của căn phòng
16         double calculateArea() {
17             // Diện tích của phòng = diện tích sàn
18             // = length * breadth
19             return length * breadth;
20         }
21
22         // Phương thức calculateVolume() tính thể tích của căn phòng
23         double calculateVolume() {
24             // Thể tích = chiều dài * chiều rộng * chiều cao
25             return length * breadth * height;
26         }
27     };
```



	Test	Expected	Got	
✓	<pre>cout&lt;&lt;"Constructor test"; Room r(20,3,4); Room *r2=new Room(10.5,5.5,5.4); delete r2;</pre>	Constructor test	Constructor test	✓
✓	<pre>//Area test Room r(5,4,5); cout &lt;&lt; r.calculateArea() &lt;&lt; endl;</pre>	20	20	✓

Passed all tests! ✓



## Câu hỏi 6

Đúng

Trong thân hàm `main()`, hãy viết chương trình chính tạo ra một mảng các đối tượng thuộc **class Course** có **n** phần tử (**n** được nhập từ người sử dụng – sử dụng kỹ thuật tạo mảng động bằng lệnh **new**). Sau đó nhập và hiển thị dữ liệu cho **n** đối tượng này bằng hai phương thức **getinfo()** và **disinfo()** đã được hiện thực sẵn.

[English]

In **main**, write a program that creates an array of Course objects with **n** elements (**n** is input from user). The array is created using dynamic allocation with operator **new**.

Then input and display data for each element of array using two implemented methods: **getinfo** and **disinfo**.

For example:

Test	Input	Result
1	2 1 2 3 4 1 3 4 5	ID: Number of Students: Number of Teachers: Number of TAs: ID: Number of Students: Number of Teachers: Number of TAs: CourseID = 1 Number of student = 2 Number of teacher = 3 Number of TA = 4  CourseID = 1 Number of student = 3 Number of teacher = 4 Number of TA = 5

Answer: (penalty regime: 0 %)

Reset answer

```

1  #include <iostream>
2  using namespace std;
3
4  // Giả sử class Course đã được định nghĩa như sau:
5  class Course {
6  private:
7      int courseID;
8      int numOfStudents;
9      int numOfTeachers;
10     int numOfTAs;
11 public:
12     void getinfo();
13     void disinfo();
14 };
15
16 // Hiện thực phương thức getinfo() để nhập thông tin khóa học
17 void Course::getinfo() {
18     cin >> courseID >> numOfStudents >> numOfTeachers >> numOfTAs;
19 }
20
21 // Hiện thực phương thức disinfo() để hiển thị thông tin khóa học
22 void Course::disinfo() {
23     cout << endl << "CourseID = " << courseID << endl;
24     cout << "Number of student = " << numOfStudents << endl;
25     cout << "Number of teacher = " << numOfTeachers << endl;
26     cout << "Number of TA = " << numOfTAs << endl;
27 }
28
29 int main() {
30     int n;
31
32     // Nhập số lượng khóa học
33     cin >> n;
34
35     // Tạo mảng động các đối tượng Course với n phần tử
36     Course* courses = new Course[n];
37
38     // Nhập thông tin cho từng khóa học
39     for (int i = 0; i < n; i++) {
40         cout << "ID: Number of Students: Number of Teachers: Number of TAs: ";

```

```

41     courses[i].getinfo();
42 }
43 // Hiển thị thông tin của tất cả các khóa học
44 for (int i = 0; i < n; i++) {
45     courses[i].disinfo();
46 }
47
48 // Giải phóng bộ nhớ đã cấp phát
49 delete[] courses;
50
51 return 0;
52 }

```

	Test	Input	Expected	Got	
✓	1	2 1 2 3 4 1 3 4 5	ID: Number of Students: Number of Teachers: Number of TAs: ID: Number of Students: Number of Teachers: Number of TAs: CourseID = 1 Number of student = 2 Number of teacher = 3 Number of TA = 4  CourseID = 1 Number of student = 3 Number of teacher = 4 Number of TA = 5	ID: Number of Students: Number of Teachers: Number of TAs: ID: Number of Students: Number of Teachers: Number of TAs: CourseID = 1 Number of student = 2 Number of teacher = 3 Number of TA = 4  CourseID = 1 Number of student = 3 Number of teacher = 4 Number of TA = 5	✓

Passed all tests! ✓

## Câu hỏi 7

Đúng

Cho định nghĩa class:

```
class ClockType
{
    public:
        void setTime(int, int, int);
        void getTime(int&, int&, int&) const;
        void printTime() const;
        clockType(int, int, int); //constructor with parameters
        clockType(); //default constructor
```

```
private:
    int hr;
    int min;
    int sec;
```

```
};
```

Cho đoạn code sau:

```
clockType myClock;
int hours;
int minutes;
int seconds;
myClock.getTime(hours, minutes, seconds);
cout << "hours = " << hours << ", minutes = " << minutes << ", seconds = " << seconds << endl;
```

Trong câu lệnh `myClock.getTime(hours, minutes, seconds);` hàm thành viên `getTime` được thực thi. Các giá trị `hr`, `min` và `sec` của `myClock` được hàm sử dụng để cài đặt giá trị của ba biến `hours`, `minutes`, `seconds` tương ứng.

**Yêu cầu:** SV hiện thực phương thức **getTime** để thực hiện được mô tả như trên.

**Lưu ý:** `hr`, `min`, `sec` cần thỏa mãn các điều kiện sau. Nếu tham số đầu vào không thỏa mãn điều kiện bên dưới thì ta gán giá trị 0 cho biến thành viên tương ứng.

- $0 \leq \text{hr} < 24$
- $0 \leq \text{min} < 60$
- $0 \leq \text{sec} < 60$

[English]

Given the class definition:

```
class ClockType
{
    public:
        void setTime(int, int, int);
        void getTime(int&, int&, int&) const;
        void printTime() const;
        clockType(int, int, int); //constructor with parameters
        clockType(); //default constructor
    private:
        int hr;
        int min;
        int sec;
};
```

And the following code fragment:

```
clockType myClock;
int hours;
int minutes;
int seconds;
myClock.getTime(hours, minutes, seconds);
cout << "hours = " << hours << ", minutes = " << minutes << ", seconds = " << seconds << endl;
```



In the statement `myClock.getTime(hours, minutes, seconds);` `getTime` member function is executed. The `hr`, `min`, and `sec` values of `myClock` object are used by the function to set the values of the three variables `hours`, `minutes`, and `seconds` respectively.

**Requirement:** Student implements the **`getTime`** method for it to do as described above.

**Note:** `hr`, `min`, `sec` need to satisfy the following conditions. If the input parameter does not satisfy the condition below, we assign the value 0 to the corresponding member variable:

- $0 \leq \text{hr} < 24$
- $0 \leq \text{min} < 60$
- $0 \leq \text{sec} < 60$

For example:

Test	Result
<pre> ClockType myClock; myClock.setTime(5, 4, 30); int hours, minutes, seconds; myClock.getTime(hours, minutes, seconds); cout &lt;&lt; "Hours = " &lt;&lt; hours &lt;&lt; ", minutes = " &lt;&lt; minutes &lt;&lt; ", seconds = " &lt;&lt; seconds &lt;&lt; endl; </pre>	Hours = 5, minutes = 4, seconds = 30
<pre> ClockType myClock(25, 61, 61); int hours, minutes, seconds; myClock.getTime(hours, minutes, seconds); cout &lt;&lt; "Hours = " &lt;&lt; hours &lt;&lt; ", minutes = " &lt;&lt; minutes &lt;&lt; ", seconds = " &lt;&lt; seconds &lt;&lt; endl; </pre>	Hours = 0, minutes = 0, seconds = 0

**Answer:** (penalty regime: 0 %)

Reset answer

```

1 class ClockType
2 {
3 public:
4     ClockType(int, int, int);
5     ClockType();
6     void printTime() const;
7     void setTime(int, int, int);
8
9     void getTime(int&, int&, int&) const;
10
11
12 private:
13     int hr;
14     int min;
15     int sec;
16 };
17
18 void ClockType::printTime() const
19 {
20     if (hr < 10)
21         cout << "0";
22     cout << hr << ":";
23     if (min < 10)
24         cout << "0";
25     cout << min << ":";
26     if (sec < 10)
27         cout << "0";
28     cout << sec;
29 }
30
31 // TODO
32 void ClockType::getTime(int& hours, int& minutes, int& seconds) const
33 {
34     // Gán giá trị của các biến thành viên cho các tham số tham chiếu
35     hours = hr;
36     minutes = min;

```

```

37     seconds = sec;
38 }
39

```

	Test	Expected	Got	
✓	<pre> ClockType myClock; myClock.setTime(5, 4, 30); int hours,minutes,seconds; myClock.getTime(hours, minutes, seconds); cout &lt;&lt; "Hours = " &lt;&lt; hours &lt;&lt; ", minutes = " &lt;&lt; minutes &lt;&lt; ", seconds = " &lt;&lt; seconds &lt;&lt; endl; </pre>	Hours = 5, minutes = 4, seconds = 30	Hours = 5, minutes = 4, seconds = 30	✓
✓	<pre> ClockType myClock(2, 2, 2); int hours,minutes,seconds; myClock.getTime(hours, minutes, seconds); cout &lt;&lt; "Hours = " &lt;&lt; hours &lt;&lt; ", minutes = " &lt;&lt; minutes &lt;&lt; ", seconds = " &lt;&lt; seconds &lt;&lt; endl; </pre>	Hours = 2, minutes = 2, seconds = 2	Hours = 2, minutes = 2, seconds = 2	✓

Passed all tests! ✓

► [Show/hide question author's solution \(Cpp\)](#)

