

#### **REVIEW FINAL EXAMS**

Đồng Quốc Thắng - 23521421 Bùi Ngọc Thiên Thanh - 23521436 Cao Lê Công Thành - 23521437

Ho Chi Minh City University Information of Technology (UIT)

December 10, 2024



### Outline

- 1 Đề 9
  - Bài 1
  - Bài 2
  - Bài 3

- 2 Đề 12
  - Bài 1
  - Bài 2
  - Bài 3



### Câu a:

Bài 1
Bài 2
Bài 3

Đề 12
Bài 1
Bài 2

- a. Phân biệt các phạm vi truy cập private, protected và public
- **Private**: Chỉ có thể được truy cập trong lớp chứa đối tượng.
- **Protected**: Có thể được truy cập trong lớp chứa và các lớp kế thừa.
- **Public**: Có thể được truy cập từ bất kỳ đâu, không bị giới hạn phạm vi.



### Câu b:

Bài 1

Bài 2

Bài 3

Đề 12

Bài 1

Bài 2

Bài 3

Đề 9

- b. Cho biết ý nghĩa và mục đích của các hàm get/set trong một lớp
- **Hàm get**: Được sử dụng để lấy giá trị của một thuộc tính.
- Hàm set: Được sử dụng để thay đổi giá trị của một thuộc tính, có thể bao gồm kiểm tra và logic xác thực. Mục đích của hàm get/set là duy trì tính toàn vẹn và bảo mật của dữ liệu.



### Đề bài:

Dè 9

Bài 1

Bài 2

Bài 3

Dè 12

Bài 1

Bài 2

Cho đoạn chương trình tính toán với lớp đối tượng ngày tháng năm (cNgay) như sau:

```
void main()
      cNgay ng1;
                                // ng1 sẽ có giá trị là ngày 1 tháng 1 năm 1
      cNgay ng2(2017, 1);
                               // ng2 sẽ có giá trị là ngày 1 tháng 1 năm 2017
      cNgay ng3(2017, 1, 7); // ng3 sẽ có giá trị là ngày 7 tháng 1 năm 2017
      cin>>ng1;
      cout << ng1;
      if(ng1 < ng2)
             cout << "Ngay 1 truoc ngay 2" << endl;
      else
             cout << "Ngay 1 khong truoc ngay 2" << endl;
```

Hãy định nghĩa lớp cNgay thích hợp để chương trình không bị lỗi biên dịch và chạy đúng. Lưu ý rằng không được chính sửa hàm main và sinh viên cần viết cả các lệnh #include thích hợp



### Class Diagram:

# Dè 9 Bài 1 Bài 2 Bài 3 Dè 12 Bài 1 Bài 2

```
cNgay
```

- -int ngay
- -int thang
- -int nam
- +cNgay(int nam=1, int thang=1, int ngay=1)
- +friend operator>>(istream& , cNgay& ): istream&
- +friend ostream<<(ostream&, cNgay&): ostream&
- + operator<(cNgay&):bool

Link code bài 2



Đề 9

Bài 3

Đề 12

Bài 1

Bài 3

### Class Diagram:

```
casi
                                   #string hoten
                                   #int sonam
                                   #int sodia
                                   #int sobuoi
                                   #int luong
                                   +virtual nhap():void
                                   +virtual xuat():void
                                   +virtual tinhluong():void
                                   +getluong():int
         casichuanotiteng
                                                                                  casinoititeng
                                                                      +int sogameshow
+tinhluong() override:void
+nhap() override:void
                                                                      +tinhluong() override:void
+xuat() override:void
                                                                      +nhap() override:void
                                                                      +xuat() override:void
```

Link code bài 3



Đề 9

Đề 12

### Câu a:

### a. Phân biệt khái niệm overload (tải chồng) và override (ghi đè).

Đặc điểm	Overload (Tải chồng)	Override (Ghi đè)	
Định nghĩa	Định nghĩa nhiều phương thức cùng tên, nhưng khác tham số.	Lớp con cung cấp triển khai lại phương thức đã có trong lớp cha.	
Số lượng và kiểu tham số	Phải khác nhau.	au. Phải giống nhau.	
Lớp áp dụng	Xảy ra trong cùng một lớp.	Xảy ra giữa lớp cha và lớp con.	
Thời gian áp dụng	Thực hiện tại thời gian biên dịch (compile time).	Thực hiện tại thời gian chạy (runtime).	
Cách thức gọi	Phương thức gọi dựa vào số lượng và kiểu tham số.	Phương thức gọi phụ thuộc vào kiểu đối tượng thực tế tại runtime.	

6/12



### Câu a:

## Bài 1 Bài 2 Bài 3 Đề 12 Bài 1 Bài 2

### Kết luận:

- Overloading: là quá trình định nghĩa nhiều phương thức có cùng tên nhưng khác về tham số trong cùng một lớp, và được xác định ở thời gian biên dịch.
- Overriding: là quá trình thay đổi hoặc ghi đè phương thức của lớp cha trong lớp con, và được xác định khi chương trình chạy, tùy thuộc vào loại đối tượng thực tế.

7/12



### Câu b:

Bài 1
Bài 2
Bài 3
Đề 12
Bài 1
Bài 2

### b. Phân biệt các kiểu kế thừa private, protected, public.

Kiểu kế thừa	Thành phần public trong lớp cha	Thành phần protected trong lớp cha	Thành phần private trong lớp cha
Public	Public	Protected	Không truy cập được
Private	Private	Private	Không truy cập được
Protected	Protected	Protected	Không truy cập được



### Câu b:

Bài 1
Bài 2
Bài 3
Dề 12
Bài 1
Bài 2

Đề 9

- Thành phần private ở lớp cha thì không truy xuất được ở lớp con
- Kế thừa public: Khi lớp con kế thừa 'public' từ lớp cha:
  - Các thành phần 'protected' của lớp cha trở thành 'protected' trong lớp con.
  - Các thành phần 'public' của lớp cha trở thành 'public' trong lớp con.
- Kế thừa private: Khi lớp con kế thừa 'private' từ lớp cha:
  - Các thành phần 'protected' và 'public' của lớp cha trở thành 'private' trong lớp con.
- Kế thừa protected: Khi lớp con kế thừa 'protected' từ lớp cha:
  - Các thành phần 'protected' và 'public' của lớp cha trở thành 'protected' trong lớp con.



### Đề bài:

```
Bài 1
Bài 2
Bài 3

Dề 12
Bài 1
Bài 2
Bài 3
```

Xây dựng lớp thời gian (giờ, phút giây) với các toán tử », « dể nhập xuất và toán tử ++ dể tăng thời gian thêm 1.



### Class Diagram:

## Bài 1 Bài 2 Bài 3 Đề 12 Bài 1

Đề 9

#### ThoiGian

- iGio: intiPhut: int
- iGiay: int
- + ThoiGian()
- + ThoiGian(int, int, int)
- + TinhGiay(): int
- + TinhLaiGio(int): void
- + operator++(): ThoiGian
- + operator<<(): ostream&
- + operator>>(): istream&

11/12



Đề 9 Bài 1 Bài 2

Đề 12

Bài 1

Bài 2

Bài 3

### Class Diagram:

```
List of Person
                                                                                                                                Person
                                                   - vector<Person> v
                                                                                                                 - Openness O

    Conscientiousness C

                                                  + void push(Person)
                                                                                                                 - Extraversion E
                                                  + Person get_person_by_index(int i)
                                                                                                                 - Agreeableness A
                                                  + Person get person by pid(intid)
                                                                                                                 - Neuroticism N
                                                                                                                 - int id
                                                                     Trait
                                                                                                                 + int get id()
                                                  # int point
     Openness
                                                                                                                 + void describe person()
                                                  + int get_point()
                                                                                                                 + void is_noticeable()
+ void output() override
                                                  + void set_point(int p)
                                                   + virtual void output()
 Conscientiousness
                                                                                                                          Neuroticism
+ void output() override
                                                                                                                     + void output() override
                                                                                 Agreeableness
                                        Extraversion
                                   + void output() override
                                                                             + void output() override
```

Link code bài 3