

# Computer Science Advanced

## LAB 1. GIỚI THIỆU VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

### Bài 1. Nhân Viên

Hãy hiện thực class **Employee** đại diện cho nhân viên tại MindX. Class này có:

- Phương thức khởi tạo nhận vào *tên nhân viên* và *vị trí làm việc*.
- Phương thức `say_hi()` in ra màn hình dòng chữ *Hi, my name is <tên nhân viên>*.
- Phương thức `tell_position()` in ra màn hình dòng chữ *I am a <vị trí làm việc>*.

Kết quả mong đợi khi sử dụng class trên như sau:

Code	Output
<pre>john = Employee("John", "Software Engineer") john.say_hi() john.tell_position()</pre>	<pre>Hi, my name is John. I am a Software Engineer.</pre>

### Bài 2. Hình Học

Hãy viết một chương trình hỗ trợ người dùng tính *chu vi* và *diện tích* của các dạng hình học cơ bản. Để đơn giản, ta chỉ hỗ trợ tính toán trên *hình chữ nhật* và *hình tròn*. Chương trình có luồng chạy như sau:

- Yêu cầu người dùng nhập tên của hình cần tính toán dưới dạng *string*
- Yêu cầu người dùng nhập các tham số tương ứng với hình đã chọn
- Tính toán và in ra chu vi và diện tích

**Yêu cầu:** Tạo một class tương ứng với mỗi hình nêu trên để hỗ trợ việc xử lý.

Kết quả mong đợi của chương trình:

Ví dụ 1	Ví dụ 2	Ví dụ 3
<pre>Shape (rectangle circle): <b>rectangle</b> Height: 2 Width: 1  =&gt; Perimeter: 6.0 =&gt; Area: 2.0</pre>	<pre>Shape (rectangle circle): <b>circle</b> Radius: 1  =&gt; Perimeter: 6.28 =&gt; Area: 3.14</pre>	<pre>Shape (rectangle circle): <b>square</b>  =&gt; Invalid!</pre>

Phần in đậm là nội dung được nhập từ người dùng.

### Bài 3. Ngày Tháng

Python hỗ trợ module **datetime** để xử lý về thời gian và ngày tháng. Tuy nhiên, để định dạng ngày và thời gian theo cách thông thường tại Việt Nam, ta cần sử dụng cú pháp tương đối phức tạp.

**Ví dụ:** Để in ra ngày tháng và thời gian hiện tại theo định dạng **DD/MM/YYYY** và **HH:MM:SS**, ta có hai cách thông dụng:

Code	Output
<pre>from datetime import datetime  now = datetime.now()  print("Method 1") print(now.strftime("%d/%m/%Y")) print(now.strftime("%H:%M:%S"))  print("\nMethod 2") print(f"{now.day:02d}/{now.month:02d}/{now.year:04d}") print(f"{now.hour:02d}:{now.minute:02d}:{now.second:02d}")</pre>	<pre>Method 1 29/06/2021 17:37:52  Method 2 29/06/2021 17:37:52</pre>

Hãy hiện thực class **CustomDate** để hỗ trợ việc định dạng trên với:

- Phương thức khởi tạo trả về đối tượng đại diện cho thời điểm khởi tạo.
- Phương thức **get\_date()** trả về một *string* là ngày tháng theo định dạng nêu trên.
- Phương thức **get\_time()** trả về một *string* là thời gian theo định dạng nêu trên.

Kết quả mong đợi khi sử dụng class trên như sau:

Code	Output
<pre>now = CustomDate() print(now.get_date()) print(now.get_time())</pre>	<pre>29/06/2021 17:37:52</pre>

## Bài 4. DateHandler

Trong lập trình, ta thường nhóm các hàm hỗ trợ thuộc một chủ đề nào đó vào một lớp, ví dụ như hỗ trợ về các phép tính toán học hay xử lý ngày tháng. Các hàm này được tổ chức thành phương thức của lớp.

Hãy hiện thực class **DateHandler** với hai phương thức lớp hỗ trợ xử lý ngày tháng như sau:

- Phương thức **format\_date(date)** nhận vào một đối tượng **datetime** và trả về một *string* là ngày tháng theo định dạng **DD/MM/YYYY**.
- Phương thức **get\_days\_between(date1, date2)** nhận vào hai đối tượng **datetime** và trả về một số nguyên là số ngày giữa hai đối tượng.

Kết quả mong đợi khi sử dụng class trên như sau:

Code	Output
<pre>from datetime import datetime  start_date = datetime(2021, 1, 1) end_date = datetime(2022, 1, 1)  print("Start:", DateHandler.format_date(start_date)) print("End:", DateHandler.format_date(end_date)) print("Days between:",       DateHandler.get_days_between(start_date, end_date))</pre>	<pre>Start: 01/01/2021 End: 01/01/2022 Days between: 365</pre>

**Gợi ý:** Để tính số ngày giữa hai đối tượng **datetime** là **date1, date2**, ta viết **(date2-date1).days**