



LAB PRACTICE № 11: BASIC OBJECT-ORIENTED PROGRAMMING

COMP1010 Introduction to Programming

Week 12

Lab Practice Submission Instructions:

- This is an individual lab practice and will typically be assigned in the laboratory (computer lab).
- Your program should work correctly on all inputs. If there are any specifications about how the program should be written (or how the output should appear), those specifications should be followed.
- Your code and functions/modules should be appropriately commented. However, try to avoid making your code overly busy (e.g., include a comment on every line).
- Variables and functions should have meaningful names, and code should be organized into functions/methods where appropriate.
- Academic honesty is required in all work you submit to be graded. You should **NOT** copy or share your code with other students to avoid plagiarism issues.
- Use the template provided to prepare your solutions.
- You should upload your file(s) to the Canvas **before the end of the laboratory session** unless the instructor gave a specified deadline.
- Late submission of lab practice without an approved extension will incur the following penalties:
 - (a) No submission during the laboratory session will incur one (1) point deduction from the total possible score of the lab practice.
 - (b) The instructor will deduct an additional point for each business day past the deadline.
 - (c) The penalty will be deducted until the maximum possible score for the lab practice reaches zero (0%) unless otherwise specified by the instructor.

Problem 1 – Tenant Information

[CMS is required] Define a class called `Apartment` that accepts two strings and two integers as input, and assigns those inputs to four instance variables, `tenant`, `block`, `floor`, and `unit`. Assign the apartment's name ("Sunshine Apartment") to a class variable called `apartment_name`. Next, assign to the variable `R1`, `R2`, `R3`, and `R4` instances of `Apartment` as follows:

Room	Tenant	Block	Floor(F)	Unit(#)
R1	Ali Baba	B	24	1103
R2	James Nguyen	A	15	1120
R3	Mario	A	2	1250
R4	Wong	C	2	1010

Display the tenant information as shown in the following sample output:

```
Welcome to Sunshine Apartment!
Ali Baba staying at Block B, 24F-#1103.
James Nguyen staying at Block A, 15F-#1120.
Mario staying at Block A, 2F-#1250.
Wong staying at Block C, 2F-#1010.
```

Problem 2 – Update the Tenant Information

[CMS is required] Modify your program in Problem 1 to assign distinct apartment name for each tenant:

- ⇒ Ali Baba: 'Phoenix Apartment'.
- ⇒ James Nguyen: 'Rainbow Apartment'.
- ⇒ Mario: 'Hanoi Apartment'.
- ⇒ Wong: 'HLT Apartment'.

In your program, you should create a new class method called `apart_name` that, when called, add the apartment name for each tenant. Display the tenant information as shown in the following sample output:

```

Welcome to 'Rainbow Apartment'!
James Nguyen staying at Block A, 15F-#1120.

Welcome to 'Hanoi Apartment'!
Mario staying at Block A, 2F-#1250.

Welcome to 'HLT Apartment'!
Wong staying at Block C, 2F-#1010.

```

Problem 3 – Course Grade Information

[CMS is **not** required] The following table stores the course assessment scores for 5 students in Group A. The total score for laboratory, midterm exam, and final exam is 80, 60, and 70, respectively.

Student No.	Laboratory (80)	Midterm Exam (60)	Final Exam (70)
1	71	53	70
2	64	60	61
3	47	35	66
4	28	37	56
5	32	31	34

Create a class called `Student` that accepts three numbers as inputs and assigns them respectively to three instance variables: `lab`, `midterm`, and `final`. Next, create the following class methods:

- (a) `cal_lab()`: to calculate the percentage of lab score (30%).
- (b) `cal_midterm()`: to calculate the percentage of midterm score (30%).
- (c) `cal_final()`: to calculate the percentage of final score (40%).
- (d) `cal_total()`: to calculate the total score (100%).
- (e) `cal_grade()`: to automatically assign a grade by using the total score.

$[A \geq 80, 70 \leq B < 80, 60 \leq C < 70, 50 \leq D < 60, F < 50]$.

Add a string method so that when you print an instance of a student, you will see the percentage of Lab, Midterm, Final, Total, and Grade. Use the code structure in Listing 1 to complete your program. Display the Course Grade information, as shown in the following sample output:

-----Course Grade-----				
Lab (30%)	Midterm (30%)	Final (40%)	Total (100%)	Grade
26.6	26.5	40.0	93.1%	A
24.0	30.0	34.9	88.9%	A
17.6	17.5	37.7	72.8%	B
10.5	18.5	32.0	61.0%	C
12.0	15.5	19.4	46.9%	F
