

國立臺灣大學

# Problem Set – Basics

Computer Programming (EE3031), F21

Jiun-Lang Huang

EE Department, National Taiwan University

# Prob. B01: Quadratic Equation Solver

- Write `quadratic_solver.py` that computes the roots for a quadratic equation. The I/O should be as follows.

```
fin2017 > python3 quadratic_solver.py
***** Welcome to Quadratic Equation Solver *****
      ax^2 + bx + c = 0
```

```
enter a: 1
```

```
enter b: 10
```

```
enter c: 2
```

```
The two roots are:
```

```
-0.2041684766872809
```

```
-9.79583152331272
```

\* bold-face  
characters are user-entered.

```
fin2017 > python3 quadratic_solver.py
```

```
***** Welcome to Quadratic Equation Solver *****
      ax^2 + bx + c = 0
```

```
enter a: 4
```

```
enter b: 3
```

```
enter c: 6
```

```
The two roots are:
```

```
(-0.3749999999999994+1.165922381636102j)
```

```
(-0.3750000000000006-1.165922381636102j)
```

```
fin2017 >
```

# Prob. B02: 固定利率本息平均攤還

- 請使用者輸入本金、年利率與攤還年數，計算每月償還金額。
- I/O should be as follows. Save your code in `cpm.py`.

```
fin2017 > python cpm.py
```

```
enter loan amount: 7000000
```

\* 本金

```
enter annual interest rate (%): 1.4
```

\* 年利率

```
enter length of term (years): 20
```

\* 攤還年數

```
monthly repay = 33457.2069721669
```

\* 每月償還金額

```
fin2017 >
```

- 每月償還金額公式

$$c = \frac{P \cdot r}{1 - (1 + r)^{-N}}$$

- $P$ : 本金， $r$ : 月利率， $N$ : 債還期數（月）， $c$ : 每月償還金額

# Prob. B03: Pretty Print

- Write a program `pretty_print.py` that asks the user to enter their name and then prints it on a banner of #'s.
- Format the banner according to the following examples.

```
fin2017 > python3 pretty_print.py
enter your name: John
#####
##### John #####
#####

fin2017 > python3 pretty_print.py
enter your name: James Bond
#####
##### James Bond #####
#####

fin2017 > python3 pretty_print.py
enter your name: Pan, Peter
#####
##### Pan, Peter #####
#####

fin2017 >
```

# Prob. B04: Simple Math Problem Generator

- Write a program `gen_problem_add.py` that prints out an addition problem and the answer.

```
fin2017 > python3 gen_problem_add.py  
95 + 34 = 129  
fin2017 > python3 gen_problem_add.py  
58 + 27 = 85  
fin2017 > python3 gen_problem_add.py  
76 + 51 = 127  
fin2017 >
```

- Both operands are integers, between 10 and 100.
- Your program is supposed to produce a different addition problem each time it is executed.  
(Hint: Use the `random` module.)
- Follow the output format of the above examples.