International Software Engineering Program
School of Engineering, Dept. of Computer Engineering
King Mongkut's Institute of Technology Ladkrabang

13006101 **Introduction to Computers and Programming**
Final Examination
13th December 2021, 9:30 – 12:30 Hrs

Instruction: This is an **opened book** exam.

Questions:

1. (6 marks) Write a Python function **find_word_positions(word,list_of_words)** to find all positions where the string **word** occurs in the list **list_of_words**. (*A word is not case-sensitive*, for example, "Computer" and "computer" are considered as the same word.) That is, the function returns a list of all positions of the **word** in the list; if **word** does not exist in the list at all, the function returns 0.

   For example, **find_word_positions("Python",["python","java", "c","PYTHON","Prolog"])** → [0,3].
   find_word_positions("iOS",["Windows","macOS","Linux"]) → 0.

2. (6 marks) Given a dictionary **popularity_scores** taken from an IEEE Spectrum ranking, define a Python function to process this dictionary and return **a dictionary** that represents the computer language ranking on the right based on the popularity scores of the computer languages on the left.

Dictionary **popularity_scores**

Computer Language	Scores
C++	99.7
C	96.7
Java	97.5
Python	100
C#	89.4
…	…

Dictionary

Ranking	Computer Language
1	Python
2	C++
3	Java
4	C
5	C#
…	…

Note: Some computer languages may have **the same scores**, in those cases **they share the same rank**.

3. (8 marks) Define a recursive Python function **count_operands_in_expr** to return the number of operands in an infix expression in which all operators are binary ones.

   Example: **count_operands_in_expr**((3,'**',4)) → 2

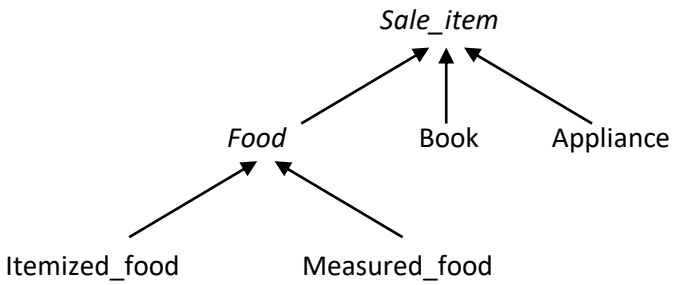
   count_operands_in_expr (((((2, '+', 4), '/', 3), '*', 2), '+', (3, '**', 4))) → 6

4. (10 marks) Define a **SavingAccount** class which has the following **properties** and **methods**:

- **Properties**
  - **bank_name**: the name of the bank who creates the account
  - **acc_name**: the name and surname of the account owner
  - **acc_id**: the account id
  - **balance**: the current balance of the account
  - **transaction_history**: transaction history of the account (stored in a list)
- **Methods**
  - **deposit(money,person,date)**: deposit **money** by **person** to the account on **date**
  - **withdraw(money,person,date)**: withdraw **money** by **person** from the account on **date**
  - **get_balance()**: get the current balance
  - **print_statement()**: print the statement according to **transaction_history**

Define a **OverDrawnAccount** as a sub-class of **SavingAccount** class. An overdrawn account is different from a saving account very little, in that a saving account cannot have a negative balance whilst an overdrawn account can have a negative balance but it does not exceed **the over drawn limit**. Please define necessary properties and methods for this sub-class.

5. (10 marks) Write a Python program using **polymorphism** to calculate cost of items you purchased from a department store. ***Sale_item*** and ***Food*** are *abstract* classes whose *concrete* classes are **Book**, **Appliance**, **Itemized_food** and **Measured_food**.



Itemized_food is priced by items and Measured_food is priced according to its weight. A price of a Book is given 15% reduction from its price printed on the cover. Food and a Book do not have VAT on top of their prices, but an Appliance has VAT 7% on top of the price.

(5a) Define the abstract classes ***Sale_item*** and ***Food***. Also define the concrete classes **Book**, **Appliance**, **Itemized_food**, and **Measured_food**.

(5b) Write the main program to calculate the total cost of a **list** of the following purchased items:

- vegetable oil 2 bottles (an Itemized_food), each bottle costs 40 Bahts
- mango 1.8 Kilograms (Measured_food), each Kilograms cost 70 Bahts
- one Python book costs 200 Bahts
- one rice cooker (an Appliance) costs 1,200 Bahts
