Practical 2

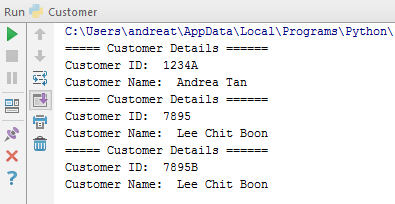
Classes, Objects and Methods

1. a) Write a Customer class that contains:

* Data attributes named \_\_customer\_id, \_\_name
* An initializer that takes parameters to set the attributes
* Accessor and mutator methods.
* A method, display\_details(), to display the customer id and name on the screen.

b) Write codes to test the Customer class.

Sample output:



c) A valid customer id should consist of 5 digits and ends with an alphabet. How to avoid an invalid customer id is assigned to customer object during initialization and through mutator method? Modify your solution.

1. Python has datetime class which you can use to obtain the year, month, day of date. The instance attributes – year, month and day return the year, month and day of the datetime object respectively. (Search on Google for the datetime API)

Write a program, CalendarApp.py, to perform 2 tasks:

* Create an instance object of datetime
* Display the current date in the format: DD-MM-YYYY

1. Given the source for Student.py

|  |
| --- |
| class Student:  def \_\_init\_\_(self, name, gender):  self.\_\_name = name  self.\_\_gender = gender  self.\_\_mark = 0.0  def get\_name(self):  return self.\_\_name  def get\_gender(self):  return self.\_\_gender  def get\_mark(self):  return self.\_\_mark  def set\_name(self, name):  self.\_\_name = name  def set\_gender(self, gender):  self.\_\_gender = gender  def set\_mark(self, mark):  self.\_\_mark = mark  def \_\_str\_\_(self):  s ='Name : {}, Gender : {} and Marks : {}'.format(self.get\_name(), self.get\_gender(), self.get\_mark())  return s  s = Student("Ms Khoo", 'F') print("=== First view ===") print(s) s.set\_mark(100) print("=== Second view ===") print(s) |

* 1. Modify the class and add in enrolment date attribute. The enrolment date is set to the current date when the student object is created. Creates an accessor method for the new attribute.
  2. Amend the \_\_str\_\_ method to include the enrolment date

Remark: What are the differences between display\_details() in question 1 and \_\_str\_\_ method in question 3?

1. A bank offers the following types of accounts to its customers, savings accounts and current accounts. Customers are allowed to deposit money into account, withdraw money, earn interest on the account. The account number should follow the following format

|  |  |
| --- | --- |
| Savings account | Current account |
| Starts with 1x-xxxxx-xx | Starts with 0x-xxxxx-xx |
| Interest rate fixed at 1% | Interest rate fixed at 0.5% |

Design your BankAccount class with

* 1. Attributes for account number, account balance
  2. Initializer with the account number as parameter and the balance is set to 0
  3. Accessor methods for account number and account balance
  4. A method, deposit that will allows user to deposit amount into the account
  5. A method, withdraw that will allows user to withdraw amount from the account, it prints a message “*You have insufficient amount to withdraw*” if withdrawal exceed the balance.
  6. A method, get\_interest\_rate that will return the interest rate of the account. e.g. 0.01 or 0.005 for 1% and 5% respectively.
  7. A method, get\_interest that will calculate the interest earned for a year
  8. Write test code that will generate the sample output given below
  9. Write codes to create a bank account. Make a transaction to deposit $500, make a transaction to withdraw $100, followed by a transaction to display the account balance, interest rate and interest earned.

Sample output:

|  |
| --- |
| Your account number is 00-12345-11 You have 500 You are left with 400 Your interest rate is 0.005 Your interest earned is 2.0 |

Remark: In this class, why it is not necessary to have a mutator method?

***-End-***