

 Programming II Diploma in CSF / IT / FI Year 1 (2019/20) Semester 2	Week 6
	60 minutes
Practical Test (15%)	

Instructions**Prior to test**

- Create a new *Console App (.Net Framework)*.
- Name the project *CustomerApp_YourStudentID*
(e.g. *CustomerApp_S12345678* if your student ID is S12345678)

Submission

- Map to **Network drive** : **\\ictspace.ict.np.edu.sg\PRG2PracticalTest**
- Upload the **WHOLE** *CustomerApp_YourStudentID* folder into the network drive (**ictspace.ict.np.edu.sg > PRG2PracticalTest > group > studentID**)

Note: It is your RESPONSIBILITY to ensure that the files are submitted correctly.

1. Analyze the class element diagram below.

Customer
-name: string -loanAmount: double -repaymentPeriod: int -interestRate: int
+Customer(string, double, int, int) +CalculateAmountDue(): double +ToString(): string

Note:

- You may assume that all customers' names are unique.
- **CalculateAmountDue()** method calculates and returns the total amount borrowed together with interest using the formula

$$\text{interest} = \text{loan amount} * \text{interest rate} * \text{repayment period}$$
 For example, if Tony loaned \$9735 at an interest rate of 2% for 3 years, the total amount due would be $9735 + 9735 * 2 / 100 * 3 = 10319.10$

Implement the **Customer** class based on the *Class Element Diagram* above.

(20 marks)

2. Add the following to the Program class:

- a. Create a List, named **customerList**, to store **Customer** objects in the Main() method.

```
class Program
{
    static void Main(string[] args)
    {
        // create customerList here
    }
}
```

(5 marks)

- b. Write an **InitCustomerList(List<Customer> cList)** method to do the following:
- Read all customer details from "loan.csv" (downloaded from MeL) and create the **Customer** objects. The details in each line correspond to "name", "loan amount", "repayment period", "interest rate" respectively.
 - Add all the **Customer** objects to **cList**.

(18 marks)

- c. Call **InitCustomerList()** method in the Main() method to populate the **customerList**.

(2 marks)

- d. Write a **DisplayOutput(List<Customer> cList)** method to display the details of all customers in the list as shown below:

Name	Loan Amount	Repayment Period	Interest Rate	Total Amount Due
=====	=====	=====	=====	=====
Tony	9,735.00	2	3	10,319.10
Bob	5,684.00	1	6	6,025.04
Henry	688.00	3	4	770.56
Clara	8,819.00	1	5	9,259.95
Dave	817.00	1	3	841.51
Esther	4,340.00	2	5	4,774.00
Fred	7,700.00	2	7	8,778.00
Gary	9,898.00	2	10	11,877.60
Ines	9,918.00	3	10	12,893.40

(10 marks)

- e. Call **DisplayOutput()** method in the Main() method to display the **customerList**.

(2 marks)

- f. Write an **AddCustomer()** method to prompt the user for the details of a new customer, create and return the **Customer** object. You are not required to do any data validation.

The user interface should look as follows (values underlined are the user input):

```
Add new customer
Customer Name: Jane
Customer loan amount: $8888
Customer repayment period: 7
Customer interest rate: 1
```

(20 marks)

- g. Call **AddCustomer()** method in the **Main()** method to create a new customer object and add it into the **customerList**. Display a message to indicate the customer is added with the name shown clearly.

(6 marks)

- h. Call **DisplayOutput()** method in the **Main()** method to display the **customerList** again.

(2 marks)

- i. Write a method **SearchCustomer(List<Customer> cList, string name)** to search for customer based on the name passed in and return the **Customer** object if it is found or **null** if it is not found.

(10 marks)

- j. In the **Main()** method, prompt the user for the name of a customer, call **SearchCustomer()** method to search for the customer and display the customer details if found or a message to indicate that the customer is not found.

Sample output (1) (value underlined is the user input):

```
Enter the name of the customer: Tony
Name: Tony   Loan Amount: 9735   Repayment Period: 2   Interest Rate: 3
```

Sample output (2) (value underlined is the user input):

```
Enter the name of the customer: Gerald
Gerald is not found!
```

(5 marks)

PLAGIARISM WARNING:

If a student is found to have submitted work not done by him/her, he/she will not be awarded any marks for this practical test. Disciplinary action may also be taken.

Similar action will be taken for student who allows other student(s) to copy his/her work, or posting any solutions or code related to the practical test before the end of the hour for the test.

*** END OF PAPER ***