中正中学(义顺)
CHUNG CHENG HIGH SCHOOL (YISHUN)

Preliminary Examination (2018)
Secondary 4 Express

Candidate Name		Register No	Class

COMPUTING
Paper 2 Practical (7155 / 02)

Date: **28 August 2018, Tuesday**
Duration: **2 hours 30 minutes**

Files provided:
CARD.py
EMPLOYEE.py
HOUSELOAN.xls
TELEPHONECHARGES.xls

READ THESE INSTRUCTIONS FIRST

Answer **all** questions.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

Programs are to be written in Python.
Save your work using the file name given in the question as and when necessary.

The number of marks is given in brackets [ ] at the end of each question or part question.
The total number of marks for this paper is 50.

Task 1 (10 m)	Task 2 (10 m)	Task 3 (10 m)	Task 4 (20 m)	Total (50 m)

At the end of the practical exam

Number of files submitted : ____ (including the given files)

Declared by candidate : _____
Signature

Verified by teacher : _____
Signature

This document consists of **7** printed pages

Setter : Katrina, Wong Shing Liu

Task 1a (4 marks)

A couple uses spreadsheet software to keep track of loans from different banks that will be repaid using a lump sum. You are required to finish setting up the spreadsheet to record the loan details.

Open the file **HOUSELOAN**. You will see the following data.

Save the file as **LOANLIST**_<your_name>_<class>_<index_number>.

	A	B	C	D	E	F	G
1	**House Loan**						
2	Bank	Loan Amount($)	Length of loan(in years)	Annual interest rate	Monthly instalment	Lump sum repayment	Affordability
3	A1	350000	25	0.015			
4	B21	400000	20	0.0175			
5	C3	385000	22	0.016			
6	B2	300000	28	0.0175			
7	C31	420000	23	0.016			
8	A11	250000	30	0.015			

1 Use an appropriate function to calculate the amount of payment made at the end of each month and use it to complete the **Monthly instalment** column. [1]

2 Use an appropriate function to calculate the amount that needs to be paid at the end of the loan period and use it to complete the **Lump sum repayment** column. [1]

3 Use a conditional statement to indicate if the loan is affordable. The loan is affordable if the monthly instalment is $1500 or less and put **YES** in the **Affordability** column. Otherwise put **NO** in the **Affordability** column. [2]

Save and close your file.

Task 1b (6 marks)

A company uses a spreadsheet to keep track of the telephone charges of some users. You are required to finish setting up the spreadsheet to record the telephone charges.

Open the file **TELEPHONECHARGES**. You will see the following data.

Save the file as **TRACKUSES**_<your_name>_<class>_<index_number>.

Telephone charges for month August

Customer ID	Minutes used	Plan type	Free minutes	Basic cost	Total cost before promotion	Cost after promotion
15A0251	158		50	$50.52		
16C0534	246		250	$25.55		
17B0240	105		100	$39.99		
15C0115	174		250	$25.55		
16A0256	117		50	$50.52		
15B0078	123		100	$39.99		
15C0158	350		250	$25.55		
16A0098	451		50	$50.52		
17B0154	372		100	$39.99		
15C1150	289		250	$25.55		

Plan Details

Plan type	Cost of each minute after free minutes	Free minutes
A	$0.30	50
B	$0.35	100
C	$0.22	250

4 The plan type can be found from the Customer ID. It is the 3rd character in the **Customer ID.**

Example: **Customer ID**: 15A0251 **Plan type** : A

Use an appropriate function to extract the plan type and use it to complete the **Plan type** column. [1]

5 Use an appropriate function to search for the **Cost of each minutes after free minutes** in the **Plan Details** table and use it to complete the **Total cost before promotion** column. [2]

[Total cost before promotion = (minutes used – free minutes) x cost of each minutes after free minutes + Basic cost]

6 Enter a formula to calculate the cost after promotion which is the total cost before promotion but is rounded down to the nearest 5 cents and use it to complete the **Cost after promotion** column. [1]

7 In cell **G16** enter a formula to find the 2nd highest **Cost after promotion**. [1]

8 In cell **G17** enter a formula to find the most popular plan type based on the free minutes. [1]

Save and close your file.

Task 2 (10 marks)

The following program accepts the ages for 15 employees in a company and prints out the oldest age and the average age. The ages are in the range 18 to 55.

```
oldest_age = 0
total_age = 0
employees = 15

for i in range (employees):
     age = int ( input("Enter the employee's age : ") )
     if age > oldest_age:
          oldest_age = age
     total_age = total_age + age

print ("The oldest age is ", oldest_age)
average_age = total_age / employees
print ("The average age is ", average_age)
```

Open the file **EMPLOYEE.py**

Save the file as **MYSTAFF**_<your_name>_<class>_<index_number>.

9 Edit your program so that it:

(a) Accepts marks for only 20 employees. [1]

(b) Prints out the youngest age as well as the oldest age. [4]

(c) Tests if the age input is between 18 and 55, and if not, asks the user for input again as necessary. [3]

Save your program.

10 Save your program as **VARIEDSTAFF**_<your_name>_<class>_<index_number>.

Edit your program so that it works for any number of employees. [2]

Save your program.

Task 3 (10 marks)

The following program should check who is eligible to apply for a child concession card for free rides on public transport using the following rules:

- age below seven years
- height greater than 0.9 metres
- not studying in a Primary School

The program calculates the number of children who are eligible for the child concession card and the number rejected. The program finishes when an age of zero or a height of zero is input. The number of children who are eligible for the child concession card and the number rejected are then printed out.

There are several syntax errors and logical errors in the program.

```
age = 0
height = float(0)
rejected = 0
accepted = 0
age = input("Please enter your age ")
height = float(input("Please enter your height "))
priSch=input("Are you studying in a primary school ? Y/N "))
while age <> 0 or height <> 0:
    if age >= 7 or priSch not in [Y, y] or height < 0.9 :
        print("You are not eligible to apply")
        if age >=7:
            print("because you are 7 years old or more")
        if priSch not in [Y, y]:
            print("because you are in a primary school")
        if height < 0.9:
            print("because your height is 0.9 metres or below")
        rejected = rejected + 1
    else:
        accepted = accepted + 1
        print("You can apply for the card")
    height = float(input("Please enter your height "))
    priSch=input("Are you studying in a primary school ? Y/N ")
    print(Accepted, "application(s) eligible to apply")
    print(Rejected, "application(s) rejected ")
```

Open the file **CARD**.py

Save the file as **MYCARD**_<your name>_<class>_<index number>

11 Identify and correct the errors in the program so that it works correctly accordingly to the rules above. [10]

Save your program.

Task 4 (20 marks)

You have been asked to write a program to calculate the average marks scored by each student over four class tests of a particular subject.

The program should allow you to:

- Enter the data in the format a, b, c, d where a, b, c, d are the marks scored for four class tests by a student. An example is 60, 70, 65, 40
- Only allow data entry of marks ranging from 0 to 100 inclusively where the passing marks are 50 and above
- Calculate the total number of passes in the class tests by a student
- Repeat this for a total of six students
- Find the average marks of a class test, rounded to the nearest whole number
- Display this on the screen after each input. Your output **must** look like this:

```
Student 1     3 pass(es)
Student 2     2 pass(es)
Student 3     3 pass(es)
Student 4     1 pass(es)
Student 5     3 pass(es)
Student 6     1 pass(es)

Average marks    59
```

12 Write your program and test that it works. [5]

Save your program as **MARKS1**_<your name>_<class>_<index number>

13 When your program is working, use the following test data to show your test results:

```
60, 40, 55, 50
70, 30, 75, 40
85, 80, 48, 82
35, 46, 45, 60
90, 45, 80, 85
68, 47, 40, 48
```

Take a screen shot of your results and save it as a bitmap
MARKSRESULTS_<your name>_<class>_<index number> [5]

14 Save your program as **MARKS2**_<your name>_<class>_<index number>
Extend your program to identify each class test that has less than 4 passes. Print out the name of the class test and the number of passes for each class test. Your output should look like this:

```
Test 2    1 pass(es)
Test 3    3 pass(es)
```

Save your program. [3]

15 Save your program as **MARKS3**_<your name>_<class>_<index number>
Extend your program to work for any number of class tests. [2]
Save your program.