

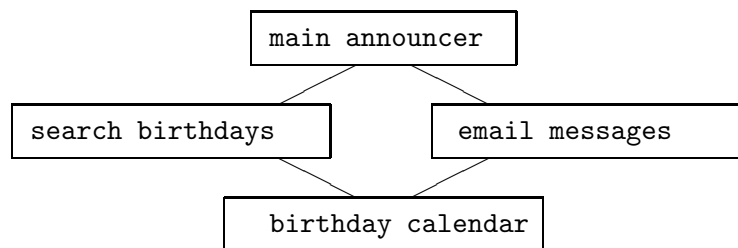
NAME: *Answers*

**Open book. No calculators, no computer.  
Write all answers on these sheets. Do not ask questions!**

question	1	2	3	4	5	6	7	total
points								
maximum	20	10	10	10	20	15	15	100

1. A manager wants an automatic birthday announcer for the employees. Each employee receives a congratulatory email on the birthday date. The manager is sent an email a week before the birthday date as a reminder to plan celebrations. The announcer program is scheduled to run automatically every day at 1 AM.
- Describe the modular design for this program. What is at the bottom of it? Draw the dependencies between the modules. For each module give a one line description of its functionality. Justify your design.

**Answer:** *At the bottom of the program are the birthday dates, names and email addresses of all employees. This data module (the birthday calendar) is imported by the search and email modules. The main program calls the search functions and then invokes the functions to send email messages.*



*One line explanations for each module:*

**main:** *calls search functions and sends emails out;*

**search:** *look into birthday calendar for matching dates;*

**email:** *defines text in the email messages;*

**calendar:** *stores birthday dates, names, and emails.*

*Justification for the design: the specific data structure for the birthday data is centralized in the data module, hidden from the main program. High cohesion is obtained by putting all search functions in one module and all the text for the emails in another module. There are as few import links as possible: low coupling. Adding extra birthdays or tailored email messages will require only local changes.*

2. Explain the differences between static whitebox and dynamic blackbox testing. Give examples of activities in both static whitebox and dynamic blackbox testing.

**Answer:** *Static whitebox testing involves code inspection, checking the source code for a list of causes for bugs. Dynamic blackbox testing is also called beta testing: we run the code without seeing its source, run it as test to fail or test to pass.*

*Static whitebox testing means that we read source code, but do not execute programs, whereas we run programs when we do dynamic blackbox testing, but we do not read the code.*

/10

3. Compare the use of open source versus proprietary software:

- (a) Why might open source software be better than proprietary software? Give two reasons, explaining each time what “better” means.

**Answer:**

1. *Open source is free of charge, available to every one.*
2. *We can modify to fix bugs or adjust to our needs.*

- (b) Why might proprietary software be better than open source software? Give two reasons, explaining each time what “better” means.

**Answer:**

1. *Proprietary software is often better supported, e.g. via a 24-hour help line.*
2. *Proprietary software may come with guarantees.*

/10

4. What is the use of the `assert` statement in Python? Describe an example of a good use of an `assert`. Justify why it is good to use an `assert` in your example.

**Answer:** *The assert statement is used to enforce conditions that must hold when the code is executed. For example: the divisor may not be zero.*

*Using assert prevents bugs from occurring and leads to code whose correctness can be proven.*

/10

5. The file `currency.txt` contains the following data:

	dollar	yen	euro
dollar	1.0	110.54	0.6838
yen	0.009046	1.0	0.006186
euro	1.4625	161.6647	1.0

From this table, we see that one euro is worth 1.4625 dollar and 161.6647 yen. In general, do not assume that the currencies are listed in the order as shown. Furthermore, there are more currencies on file than shown in the example.

Think of a program that will prompt the user for the name of a file (e.g.: `currency.txt`) and two names of currencies, say A and B. The program will then print that one A is worth  $x$  B, where the number  $x$  is retrieved from the file.

- (a) Describe in plain English (use complete sentences) the structure of the program. What data structures do you use? Which built-in functions do you apply?

**Answer:** *The lines in file are first read as strings, then splitted into lists, using the built-in function `split` applied to strings.*

*On lists, we use the built-in function `index` to locate which lines will give the correct currency conversion: with `index` we get the correct row and column from the table.*

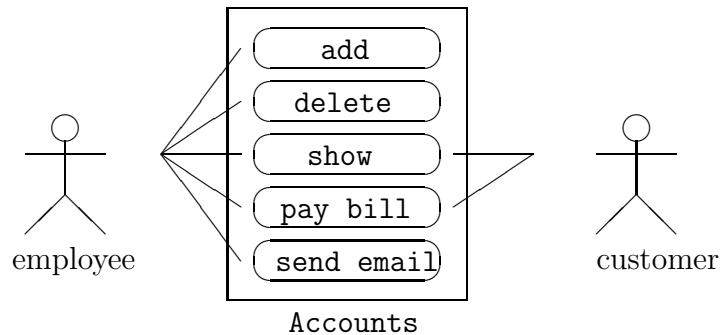
- (b) Give Python code. Do not worry about wrong user input or incorrect data.

**Answer:**

```
file = open('currency.txt','r')
A = raw_input('first currency : ')
B = raw_input('second currency : ')
s = file.readline()
L = s.split()
a = L.index(A) + 1
b = L.index(B) + 1
for k in range(0,a):
    s = file.readline()
    L = s.split()
print 'one ' + A + ' is worth ' + L[b] + ' ' + B
file.close()
```

6. Consider an account management program for a utility company. Customers can login, check the status of their account, make a payment, and logout. After logging in, employees use the program to email bills to customers.
- Use UML to draw a use case diagram for an object oriented design of the program. For each object in the diagram, write one line to define what the object represents.

**Answer:** *Employees and customers differ in their use of the accounts:*



*The class **Accounts** is a list of objects of the class **Account**. **Employees** and **customers** are instances of the class **Person**. The class **People** is a list of objects of the class **Person**. A similar use case diagram can be drawn for use of class **People**. **Employees** and **customers** may both login and logout, whereas employees can only create and delete new **Person** objects.*

/15

7. Consider the search in a phone directory. We search the directory in both ways. Either we provide a name and then the program shows the corresponding phone number, or we give a phone number and then the program shows the corresponding name.
- Draw the layout of the GUI for this search program. Indicate on your drawing what widgets you will use.

**Answer:**



*The GUI consists of two Label widgets with text **name** and **phone number**. Underneath each Label widget is an Entry widget for the user to enter a name and a phone number respectively. The bottom row has two Button widgets that respectively trigger the search for a phone number or a name.*

/15