2020

Free Python Course

O. Elijah

onlinepythonclass@gmail.com

O. ELIJAH (MSc, Ph.D)

1 PROJECT – 1

In this chapter, we shall take a look at the basic steps in developing a simple project in in python. This cover the system development

1.1 Review of Chapter 7

In chapter 7 we looked at working with files and data visualization. We practiced how to read from text file and excel files. We also worked on how to visualize data in excel.

1.2 System Development

A variety of life cycle models exist, but they all include the same constituent parts:

- a requirements-gathering phase,
- a design phase,
- a construction or implementation phase,
- a testing phase

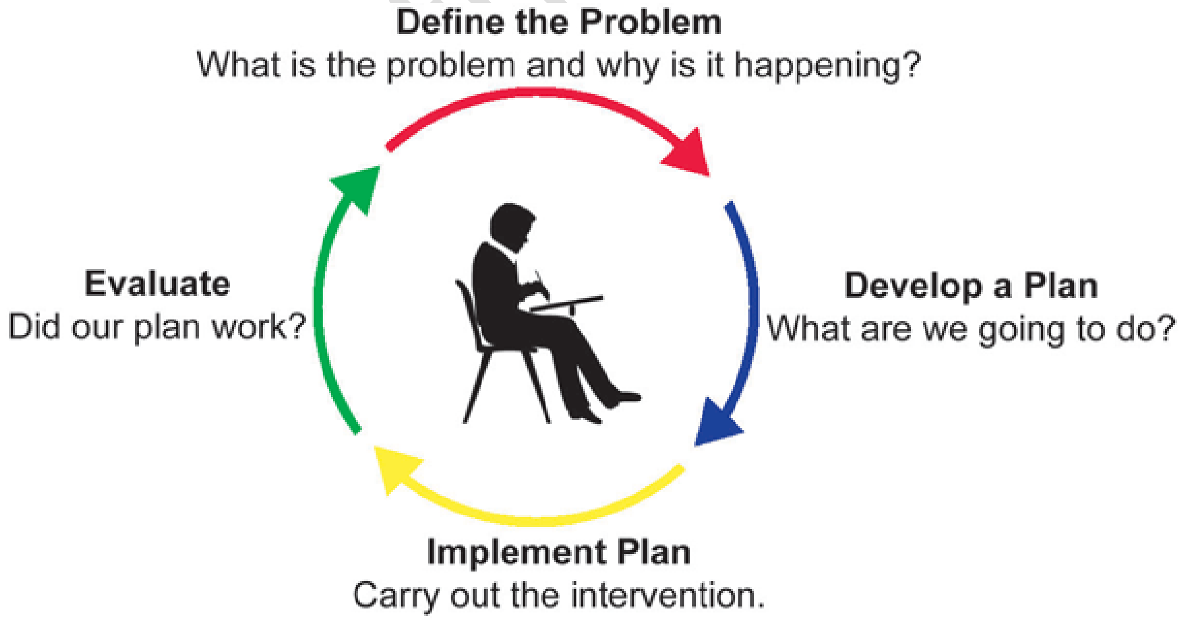


Figure 8.1 Basic component of software development cycle

Example of software development cycle is the SDLC also known as Classic Life Cycle Model (or) Linear Sequential Model (or) Waterfall Method

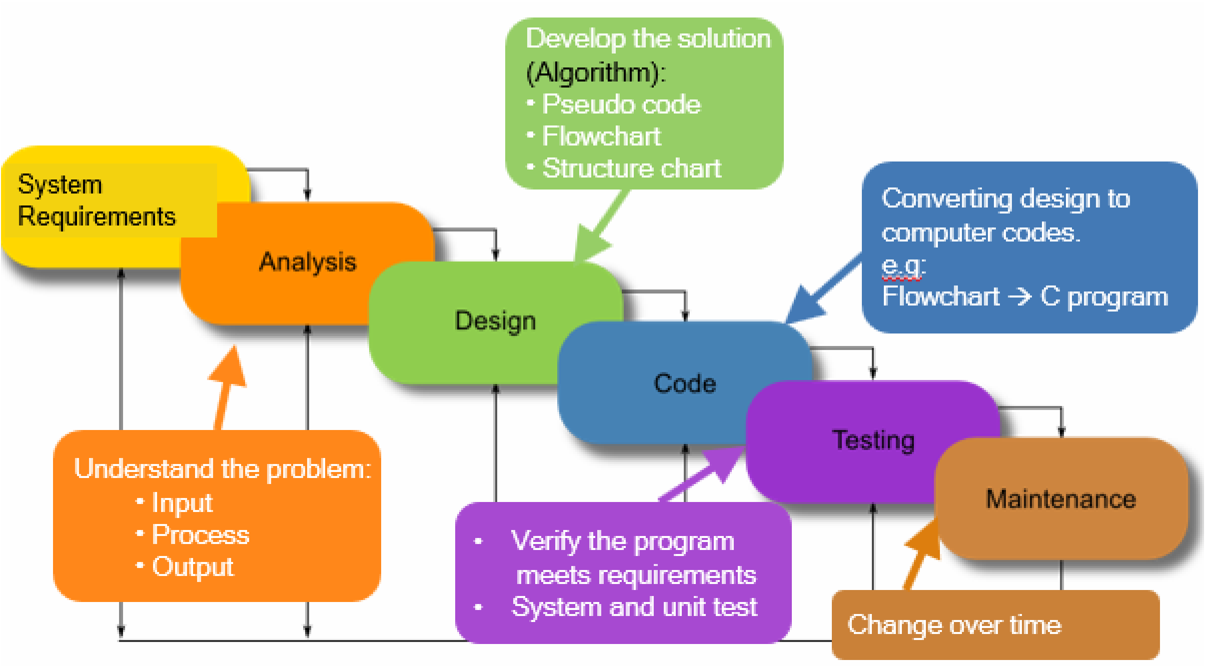


Figure 8.2. Classical life cycle model

1.3 Approach to solving a program

First you need to know the problem you intend to solve, then carry out systems requirements and analysis. In the analysis stage you should be able to define your input parameters, the process to be performed and also the output parameters.

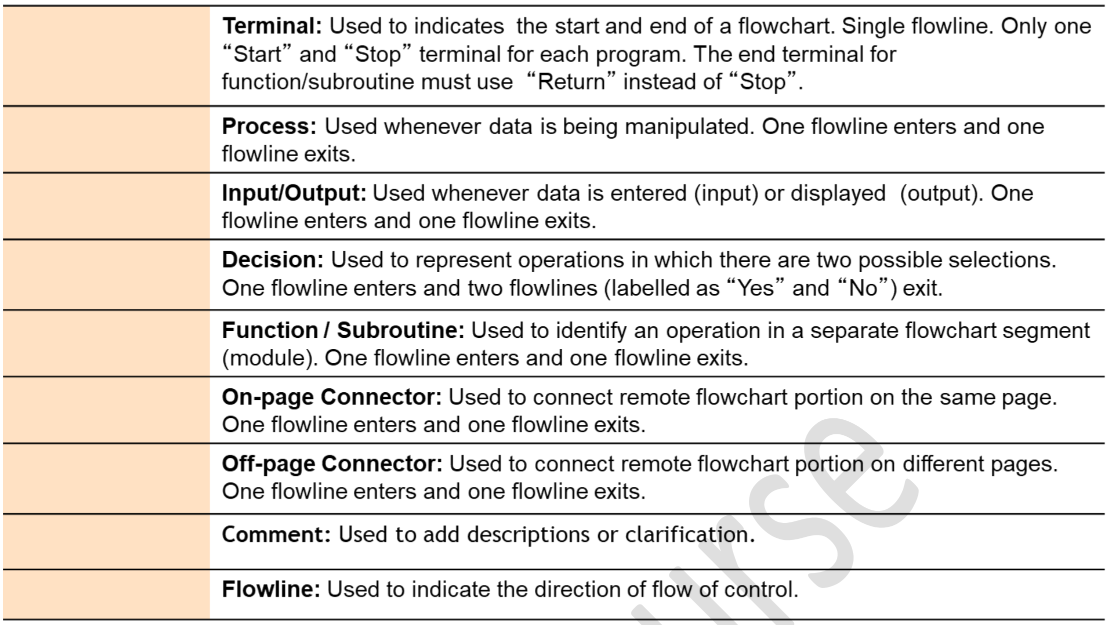
Second, you need to formulate a design of the solution and this can be in the form of algorithm. Three tools in the problem solving techniques which include

1. Development algorithms
2. Topdown designs

Note aalgorithm is a procedure for solving a problem in terms of the **actions** to be executed, and the **order** in which these actions are to be executed.

Development algorithms

- Pseudo-code - **Pseudo-code** is an artificial and informal English-like language with limited vocabulary that helps us to develop and describe algorithms.
- Flowchart – A flowchart is a graph of geometrical shapes and connected by lines to develop and represent an algorithm



	Terminal: Used to indicates the start and end of a flowchart. Single flowline. Only one "Start" and "Stop" terminal for each program. The end terminal for function/subroutine must use "Return" instead of "Stop".
	Process: Used whenever data is being manipulated. One flowline enters and one flowline exits.
	Input/Output: Used whenever data is entered (input) or displayed (output). One flowline enters and one flowline exits.
	Decision: Used to represent operations in which there are two possible selections. One flowline enters and two flowlines (labelled as "Yes" and "No") exit.
	Function / Subroutine: Used to identify an operation in a separate flowchart segment (module). One flowline enters and one flowline exits.
	On-page Connector: Used to connect remote flowchart portion on the same page. One flowline enters and one flowline exits.
	Off-page Connector: Used to connect remote flowchart portion on different pages. One flowline enters and one flowline exits.
	Comment: Used to add descriptions or clarification.
	Flowline: Used to indicate the direction of flow of control.

Top-down design

- Structured chart - Illustrates the partitioning of a problem into sub-problems and shows the hierarchical relationships among the parts.
- It is a tool to aid in software design - aid the programmer in *dividing and conquering* a large software problem, that is, recursively breaking a problem down into parts that are small enough to be understood by a human brain.
- The process is called *top-down design*, or *functional decomposition*

Structured software follows rules:

① Modules are arranged hierarchically.

② There is only one root (i.e., top level) module.

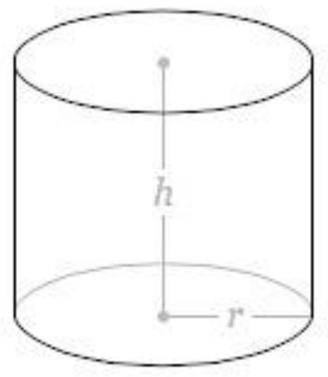
③ Execution begins with the root module.

④ Program control must enter a module at its entry point and leave at its exit point.

⑤ Control returns to the calling module when the lower level module completes execution.

1.4 Simple project

Write the pseudo-code program and a flowchart to calculate and display the volume of a cylinder.



Solution 1: pseudo-code

```
1. Get the high, let say h
2. Get the radius, let say r
3. Calculate the result let say V, $V = \pi * h * r^2$
4. Display the result, V
```

Solution 2: flowchart

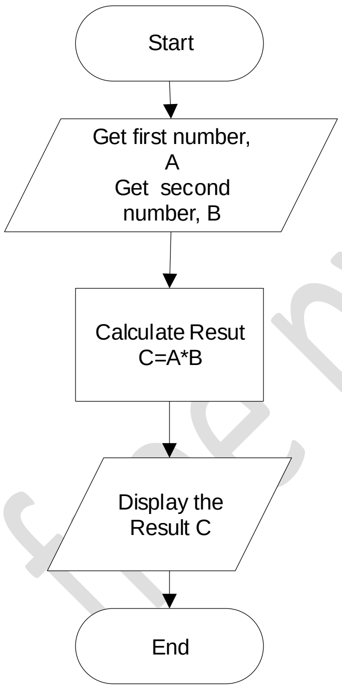


Figure 8.3 Example of topdown designs of the operations of an ATM

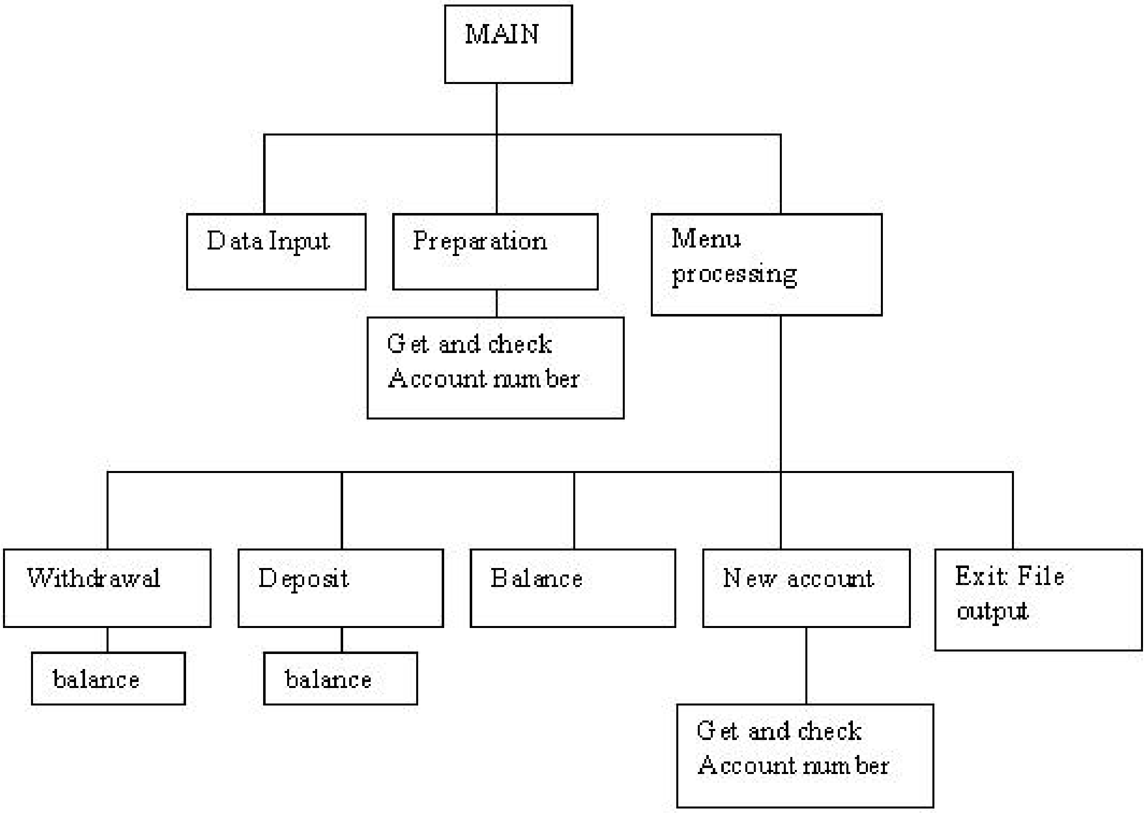


Figure 8.4 : Example of ATM Structure chart

1.5 Application

You have been hired under a student working scheme program. Your first task is to write a python program to assist in one of the professors in the faculty to grade the final exam of his/her students. The exam consists of 20 multiple-choice questions. Each question has one of four possible answers: A, B, C, or D. The program will read in the students' answers and the correct answers from files and prints out the result onto the screen and also the file output. There are at least 15 students in class.

Input

The students' answers are stored in a data file named "StudentAnswers.dat" as shown in Figure 8.5. In this file, the first column represents the students' name (1 word), second column represent the students' ID and the third column until the last column represent the students' answer of question 1 until question 20. In this example (Figure 8.5), there are only five students. The correct answers for all the questions are stored in a excel file named "CorrectAnswers.xlsx" as shown in Figure 8.6.

Output

The program needs to determine and print out the following items **on the screen** as shown in Figure 8.7:

- Ask the user to enter student's ID
- Display the student's ID and student's name
- Compare the student's answers and the correct answers. Calculate and display the total number of question missed by the student.
- Display the list of the questions missed by the students, showing the correct answers and the incorrect answers by the student for each missed question.
- Calculate and display the percentage of questions answered correctly. This can be calculated as:

$$percentage = \frac{correctly\ answered\ questions}{total\ number\ of\ questions} \times 100$$

- Display the grade of the students based on the percentage as follows:
    - $80 \leq$ percentage ≤ 100 - grade is A
    - $70 \leq$ percentage < 80 - grade is B
    - $60 \leq$ percentage < 70 - grade is C
    - percentage < 60 – grade is F

The program needs to print out the following item onto the **file output** as shown in Figure 8.8:

- Display all students' name, students' ID, students' percentage and students' grade

- The number of student is based on the input data from file "StudentAnswers.xlsx".
- Use array (one-dimension or two-dimension) to store the input data from file and the output data.
- The program should be written in several user-defined functions for example `readFile()` function to read input data, `compareAnswer()` function to check the student's answers, `printMissQuestion()` to display the missed questions and the correct answer, `printReport()` to display output onto the output file etc. Each function must be implemented with the concept of parameter passing. Use appropriate arguments for each function. Do not use global variables.

```
Abdullah    A19EE0180   A B A D A C C D A B C D A D C D A B C D

LuDong      A19EE0160   A B D D A C B D D B C A A D C D A B C D

Syarifah    AC12CS678   A A D C B C D D A B C C A A C D A D C D

Sivarajah   AC12CS123   C B C B A C C D B B C C A B C D C C C D

Wendy       B19EE0167   A C C D A B C C A A C D A A A D A B C D
```

Figure 8.5: file “StudentAnswers”

```
A B C D A B C D A B C D A B C D A B C D
```

Figure 8.6: file “CorrectAnswers”

```
Enter the student ID: B19EE0167

EXAM RESULT
Name              : Wendy
Student ID  : B19EE0167
Number of questions missed: 5
List of the questions missed:
Question        Correct Answer      Student Answer
2                 B                       C
8                 D                       C
10                B                       A
14                B                       A
15                C                       A
Percentage: 75% , GRED : B
```

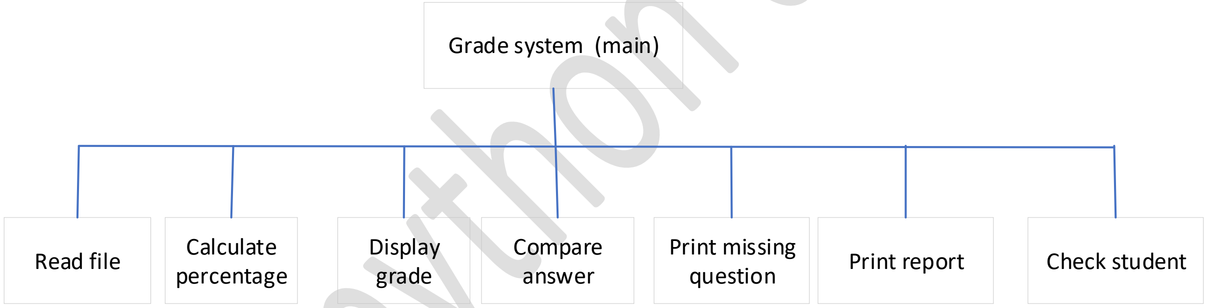
Figure 8.7: example of output on the screen

```
LIST OF STUDENTS AND GRADES

NAME          ID             PERCENTAGE
        GRADE

Abdullah     A19EE0180    85             A

LuDong       A19EE0160    70             B

Syarifah     AC12CS678    55             F

Sivarajah    AC12CS123    65             C

Wendy        B19EE0167    75             B
```

Figure 8.8: example of output in the output file.

Solution:

First stage, we need to do our topdown design. Here is a simple topdown design of our software solution



Top down design

The **main** program calls the functions

The **read file** function loads the data from the excel file into the python environment

The **calculate percentage** computes the grade of the student

The **compare function** compares the students answers with the correct answers

The **print missing question** should display the list of missed questions in a tabular format

The **print report** should display the list of students and their grade

The **check student** should validate if a student matric number is on the excel file

The **display grade** prints the percentage and grade of the student

Second stage,

First you need to import the excel python library

```
import openpyxl
```

Now we need to implement the codes for the functions. To do this you need to create the functions first

`readFile(filename):`

```
# function to read excel sheet file
def readFile(filename):
    wb_obj = openpyxl.load_workbook(filename)
    return wb_obj
```

`compareAnswer(cor_ans, stu_ans)`

```
# fucntion to compare student answer with the correct answer
def compareAnswer(cor_ans, stu_ans):
    count_missed = 0
    if len(cor_ans) == len(stu_ans):
        for i in range(len(cor_ans)):
            if cor_ans[i] == stu_ans[i]:
             # do something
                 pass
            else:
           # do some other thing
                 count_missed = count_missed + 1
    return count_missed
```

`cal_percentage(coun)`

```
# function to calculate percentage
def cal_percentage(coun):
    per_count = ((20- coun)/20)* 100
    return per_count
```

Note: The codes for the printMissQuestion has not been added to the function, hence you use the key word pass

`printMissQuestion()`

```
# function to print missing question
def printMissQuestion():
    pass
```

Note: The codes for the printReport has not been added to the function, hence you use the key word pass

`printReport():`

```
# function to print student grade
def printReport():
    pass
```

`display_grade(grad):`

```
def display_grade(grad):
    if grad >= 80:
        grade = "A"
    elif (grad >=70) & (grad <80):
        grade = "B"
    elif (grad >=60) & (grad < 70):
        grade = "C"
    else:
        grade = "F"

    return grade
```

`checkStudent(matricno, r)`

```
def checkStudent(matricno, r):
    scores = []
    student_name = 0
    matric_num = 0
    for i in range(1, m_row + 1):
        cell_obj = sheet_studa_ans.cell(row = i, column = 2)

        if cell_obj.value == matricno:
            matric_num = matricno
            student_name = sheet_studa_ans.cell(row = i, column = 1).value
            for j in range(3, sheet_studa_ans.max_column + 1):
                sco = sheet_studa_ans.cell(row = i, column = j).value
                scores.append(sco)
            break
        else:
            matric_num = "invalid"

    return [scores, student_name, matric_num]
```

Main program

```
# main program
student_answer = "StudentAnswer.xlsx"
stu_ans = readFile(student_answer)
sheet_studa_ans = stu_ans.active


correct_answer = "CorrectAnswers.xlsx" # read correct answers
corr_ans = readFile(correct_answer)
sheet_corr_ans = corr_ans.active
answers = []  # create an empty list to store the answers
for j in range(1, sheet_corr_ans.max_column + 1):
                ans = sheet_corr_ans.cell(row = 1, column = j).value
                answers.append(ans)


student_ID = input("please enter student ID: ")

m_row = sheet_studa_ans.max_row

s = checkStudent(student_ID, m_row)
#print(s[0], "  ", s[1], "  ", s[2] )

if s[2] == "invalid":
    print("Student ID is invalid")

else:
    count_miss = compareAnswer(answers, s[0])
    percentage = cal_percentage(count_miss)
    stu_grade = display_grade(percentage)

    print("Exam Result" )
    print("Name         :",s[1])
    print("Student ID   :",s[2])
    print("Number of questions missed:", count_miss)
    print("Percentage: ",percentage,"%", "  GRED :" ,  stu_grade)
```

Results

Lets check the grade for A19EE0180

```
please enter student ID: A19EE0180
Exam Result
Name          : Abdullah
Student ID    : A19EE0180
Number of questions missed: 3
Percentage:  85.0 %   GRED : A
```

Lets check the grade for B19EE0167

```
please enter student ID: B19EE0167
Exam Result
Name          : Wendy
Student ID    : B19EE0167
Number of questions missed: 5
Percentage:  75.0 %   GRED : B
```

Lets check for the grade of the student not on the list

```
please enter student ID: a1907888
Student ID is invalid
```

The complete code is the python file named week8.py and can be downloaded from the WEEK 8 folder. To run the code make sure the two excel files StudentAnswers.xlsx, CorrectAnswers.xlsx and the week8.py are in the same folder.

1.6 Reference

[1] https://plan.io/blog/software-development-process/

[2] https://www.tutorialspoint.com/python/python_functions.htm

1.7 Exercise

1. Complete the code for the printMissQuestion() and adjust the code to display the table as shown in Fig. 8.7

```
Enter the student ID: B19EE0167

EXAM RESULT
Name                 : Wendy
Student ID  : B19EE0167
Number of questions missed: 5
List of the questions missed:
Question         Correct Answer      Student Answer
2                  B                       C
8                  D                       C
10                 B                       A
14                 B                       A
15                 C                       A
Percentage: 75% , GRED : B
```

Figure 8.7: example of output on the screen

2. Complete the code for the printReport() and adjust the main code to display the report in Fig. 8.8

```
LIST OF STUDENTS AND GRADES

NAME         ID             PERCENTAGE
      GRADE

Abdullah     A19EE0180    85             A

LuDong       A19EE0160    70             B

Syarifah     AC12CS678    55             F

Sivarajah    AC12CS123    65             C

Wendy        B19EE0167    75             B
```

Figure 8.8: example of output in the output file.

Exercise is due on the 11Th of March