Objectives for class 11

--- Chapter 6 ---

- 6.6 To invoke a function using positional arguments or keyword arguments (§6.5)
- 6.7 To invoke functions defined from another program (§6.7)
- 6.8 To apply the concept of function in software development and design (§6.13)

Use **positional** arguments to call function

- Positional arguments match parameters in *order*, *number*, and *compatible type*

Function Definition of nPrintln

parameters

```
def nPrintln(message, n):
    for i in range(0, n):
        print(message)
```

Function Calls of nPrintln

```
nPrintln(“CIS”,3) # Correct!

nPrintln(3,”CIS”)
#Wrong! Order not match

nPrintln(“CIS”,3,4)
#Wrong! Number not match

nPrintln(“CIS”,”3”)
#Wrong! Type not match
```

Use **keyword** arguments to call function

- parameterName=value
- Keyword arguments match parameters in *number* and *compatible type, not in order*

Function Definition of nPrintln

parameters

```
def nPrintln(message, n):
    for i in range(0, n):
        print(message)
```

Function Calls of nPrintln

```
nPrintln(message=“CIS”,n=3)
# Correct!

nPrintln(n=3,message=”CIS”)
# Correct!

nPrintln(message=“CIS”,n=3,4)
# Wrong! Number not match

nPrintln(message=“CIS”,n=”3”)
# Wrong! Type not match
```

Use both positional and keyword arguments to call function

- Positional arguments first, then keywords arguments

Function Definition of nPrintln

parameters

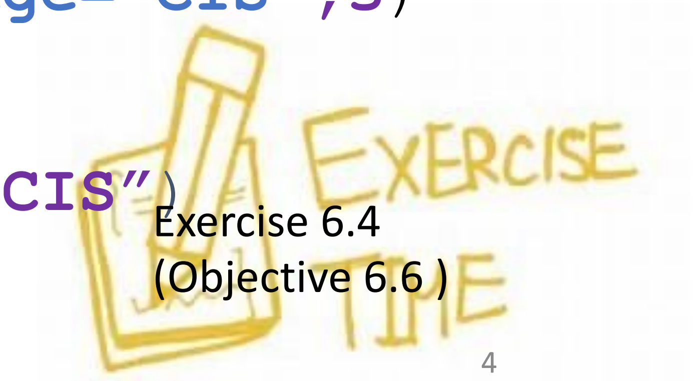
```
def nPrintln(message, n):
    for i in range(0, n):
        print(message)
```

Function Calls of nPrintln

```
nPrintln(“CIS”,n=3)
# Correct!

nPrintln(message=“CIS”,3)
# Wrong!

nPrintln(n=3,”CIS”)
# Wrong!
```



Exercise 6.4
(Objective 6.6)

How to use a function defined in another program?

PrintHello.py

```
def print1hello():
    print("Hello World")

def print2hello():
    print("Hello World")
    print("Hello World")

def print100hello():
    for i in range(0,100):
        print("Hello World")
```

How to print "Hello World" once using **print1hello()** from PrintHello.py ?

Method 1: import whole program

Method 2: import only the function

Import whole program

PrintHello.py

```python
def print1hello():
    print("Hello World")

def print2hello():
    print("Hello World")
    print("Hello World")

def print100hello():
    for i in range(0,100):
        print("Hello World")
```

TestPrint1.py

```python
import PrintHello


PrintHello.print1hello()
```

- Like we import **math** library
- **Import** whole python file (**module**)
- Access function imported using a **dot**

Import only one function from another program

PrintHello.py

```
def print1hello():
    print("Hello World")

def print2hello():
    print("Hello World")
    print("Hello World")

def print100hello():
    for i in range(0,100):
        print("Hello World")
```

TestPrint2.py

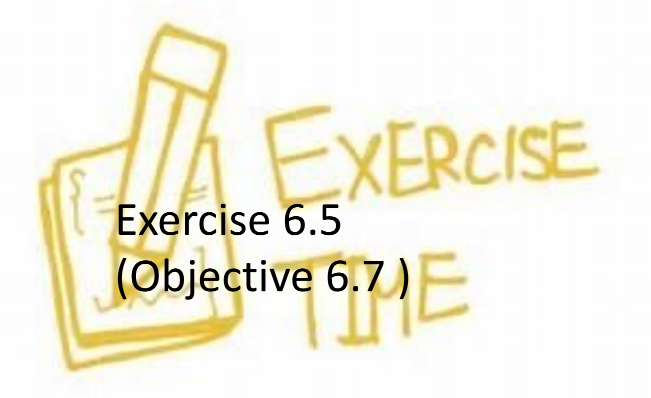
```
from PrintHello import print1hello

print1hello()
```

- **Import** only one function **from** the python file (**module**)
- **Dot** not needed

Some notes about user defined functions

- The python/module file should be placed in the same directory with your other programs.
- A python program can contain more than one functions.
- Functions are with different names in the same program.



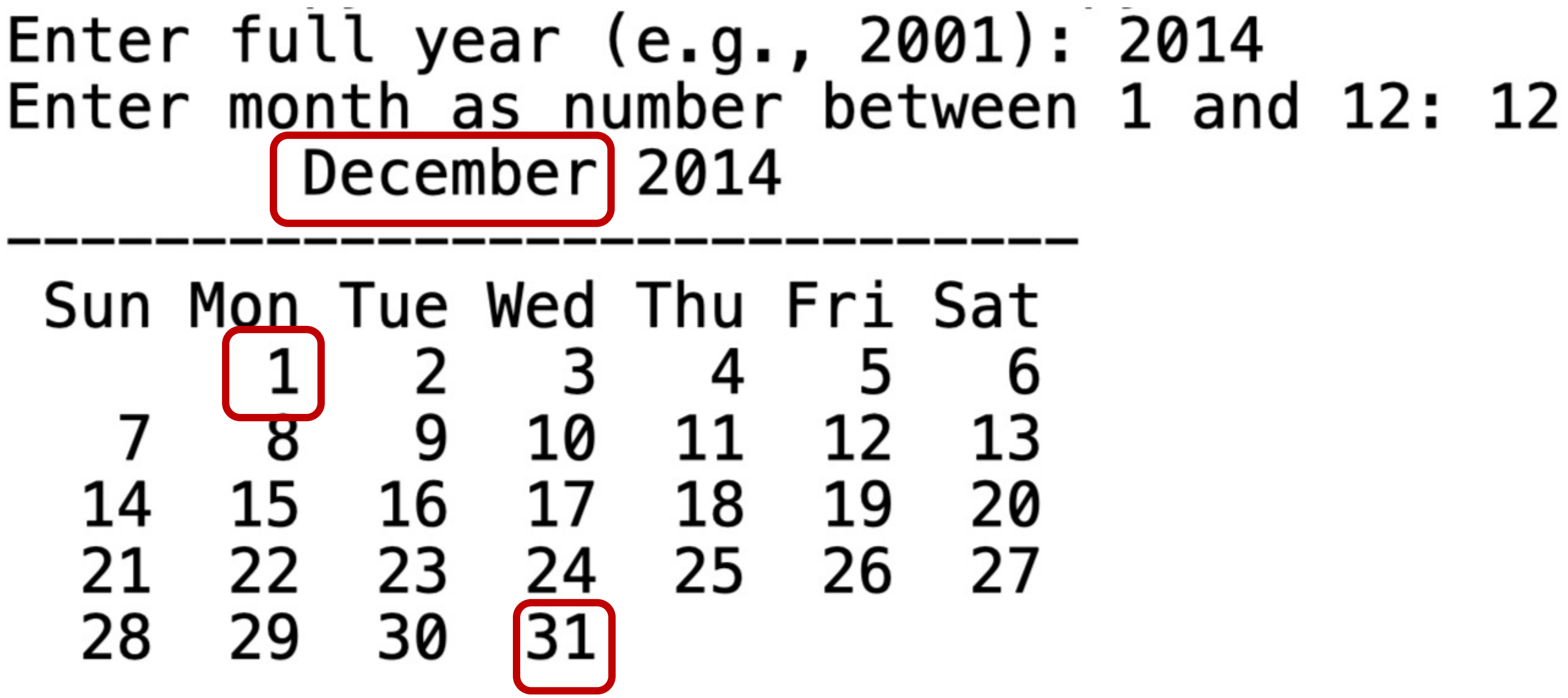
Exercise 6.5
(Objective 6.7)

Benefits of using function in software development

- **Promote code reuse**
- **Information hiding**
  - Implementations encapsulated and hidden from the client that uses functions.
- **Divide and conquer**
  - Break down problem into manageable subproblems.
- Easier Debugging
- Better facilitating teamwork

Software Development with Functions -- PrintCalender Case Study

- Let us use the PrintCalendar example to demonstrate the stepwise refinement approach.



```
Enter full year (e.g., 2001): 2014
Enter month as number between 1 and 12: 12
        December 2014
-----------------------------
 Sun Mon Tue Wed Thu Fri Sat
       1   2   3   4   5   6
   7   8   9  10  11  12  13
  14  15  16  17  18  19  20
  21  22  23  24  25  26  27
  28  29  30  31
```

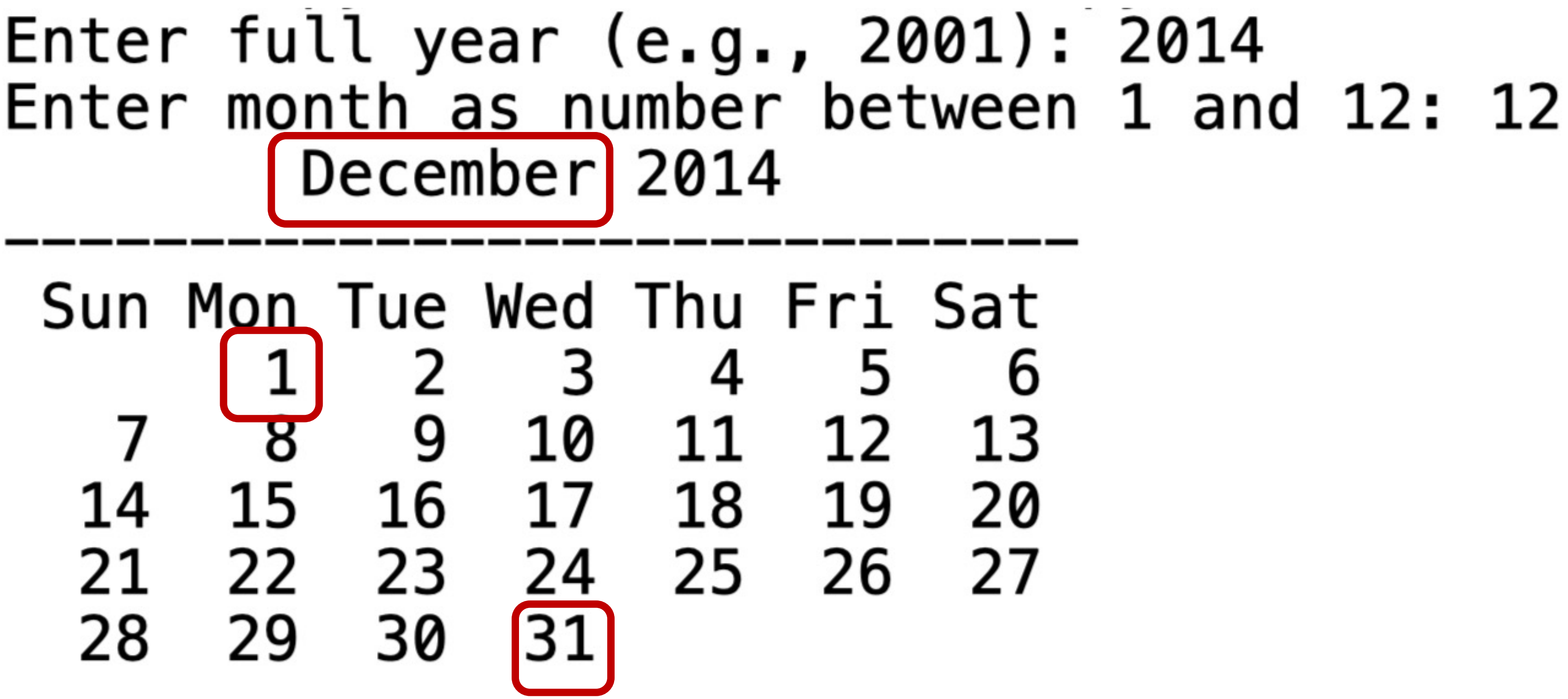
Think about Challenges

Read input

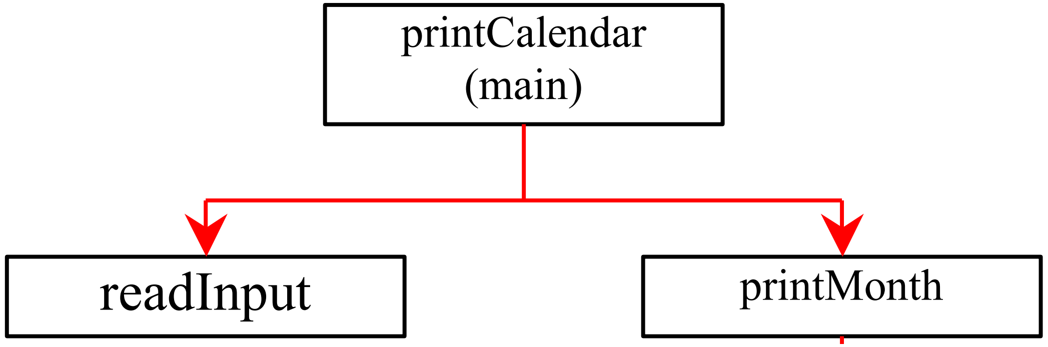
Print Month Name

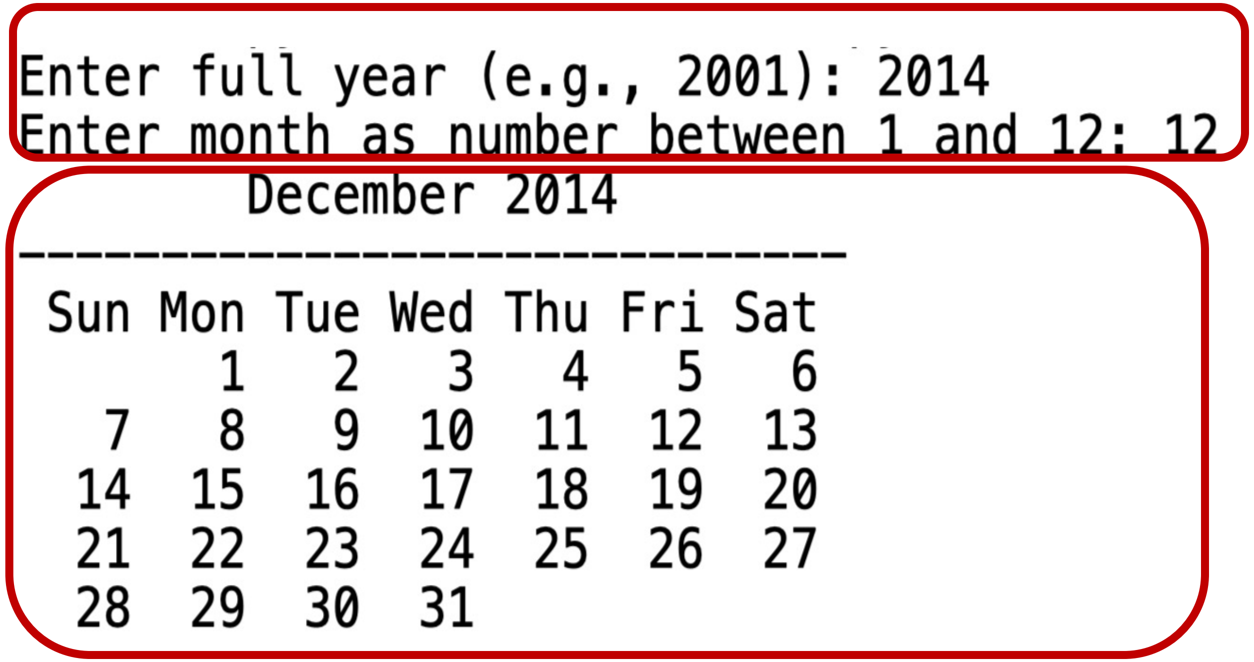
Get Start Day

Get Number of Days in the month



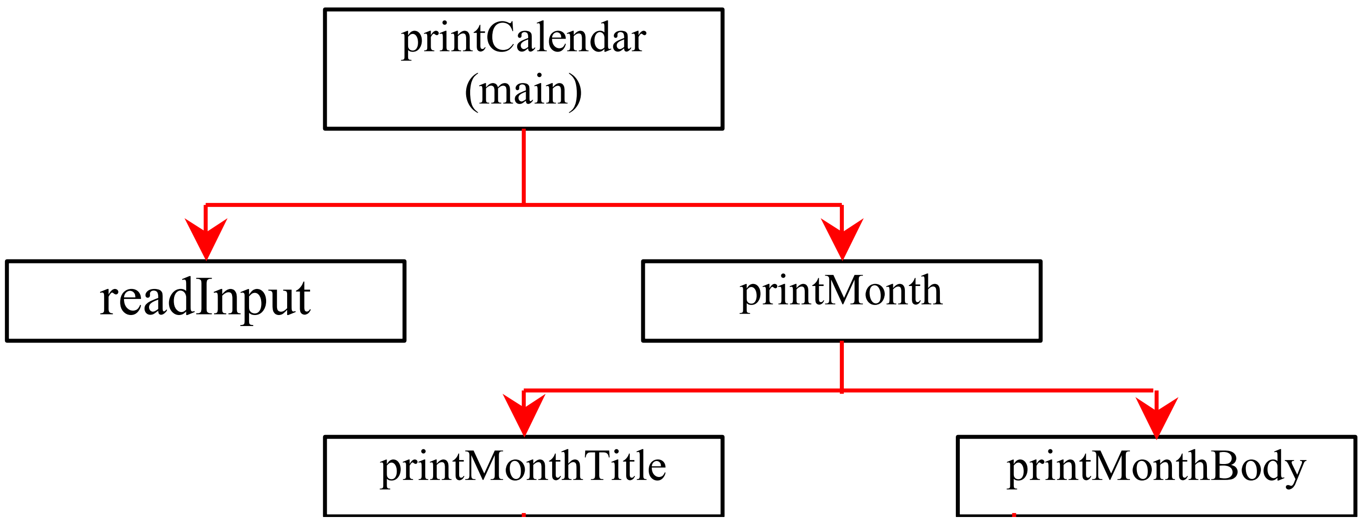
```
Enter full year (e.g., 2001): 2014
Enter month as number between 1 and 12: 12
        December 2014
-----------------------------
 Sun Mon Tue Wed Thu Fri Sat
       1   2   3   4   5   6
   7   8   9  10  11  12  13
  14  15  16  17  18  19  20
  21  22  23  24  25  26  27
  28  29  30  31
```

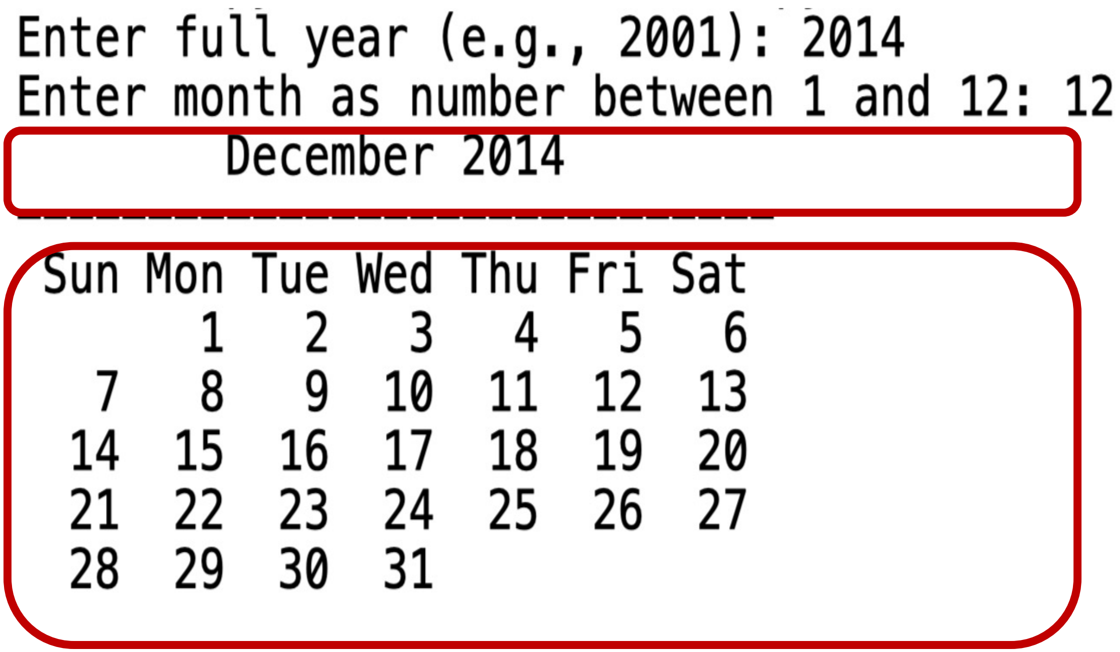




```
Enter full year (e.g., 2001): 2014
Enter month as number between 1 and 12: 12
        December 2014
-----------------------------
 Sun Mon Tue Wed Thu Fri Sat
       1   2   3   4   5   6
   7   8   9  10  11  12  13
  14  15  16  17  18  19  20
  21  22  23  24  25  26  27
  28  29  30  31
```

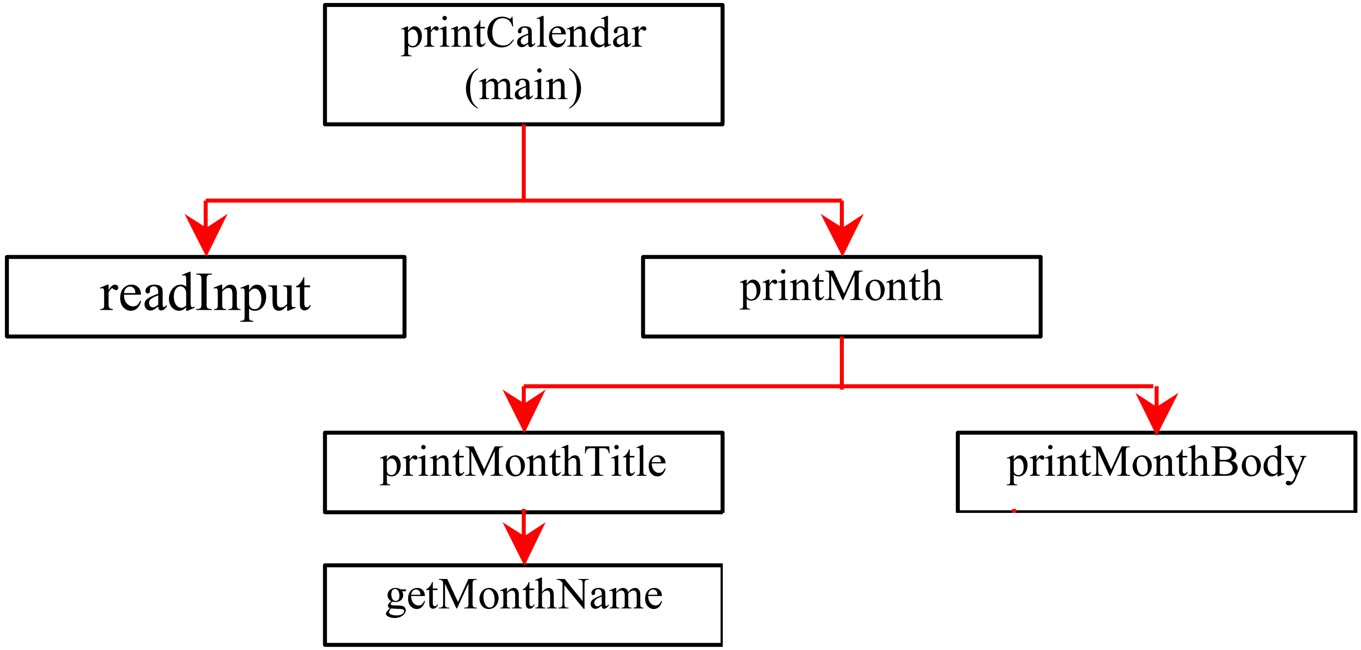
System Design Diagram

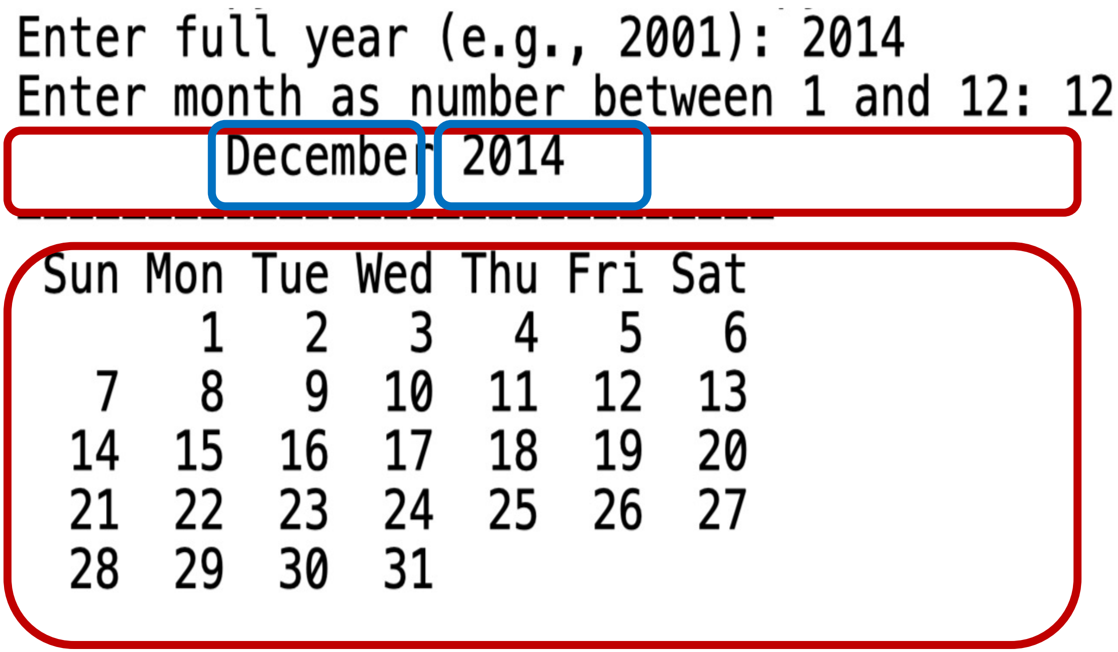




```
Enter full year (e.g., 2001): 2014
Enter month as number between 1 and 12: 12
        December 2014
-----------------------------
 Sun Mon Tue Wed Thu Fri Sat
       1   2   3   4   5   6
   7   8   9  10  11  12  13
  14  15  16  17  18  19  20
  21  22  23  24  25  26  27
  28  29  30  31
```

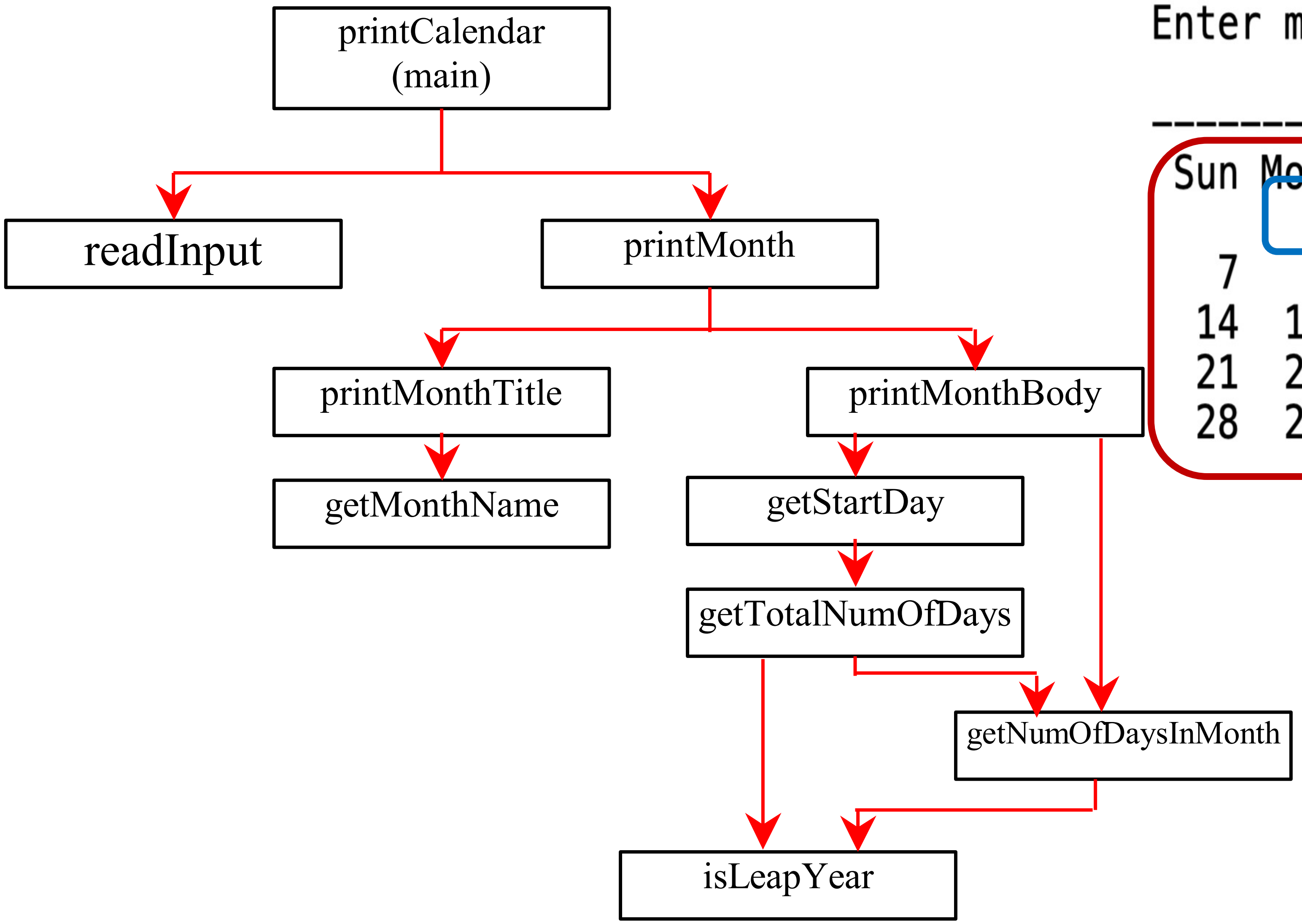
System Design Diagram





```
Enter full year (e.g., 2001): 2014
Enter month as number between 1 and 12: 12
        December 2014
-----------------------------
 Sun Mon Tue Wed Thu Fri Sat
       1   2   3   4   5   6
   7   8   9  10  11  12  13
  14  15  16  17  18  19  20
  21  22  23  24  25  26  27
  28  29  30  31
```

System Design Diagram



printCalendar (main)

readInput

printMonth

printMonthTitle

printMonthBody

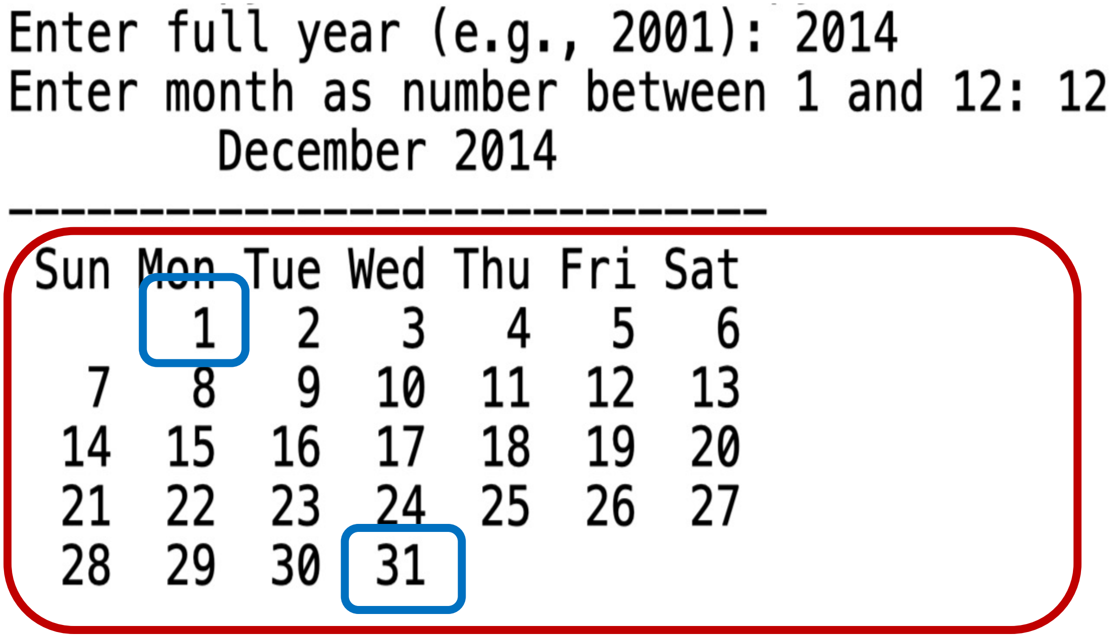
getMonthName

getStartDay

getTotalNumOfDays

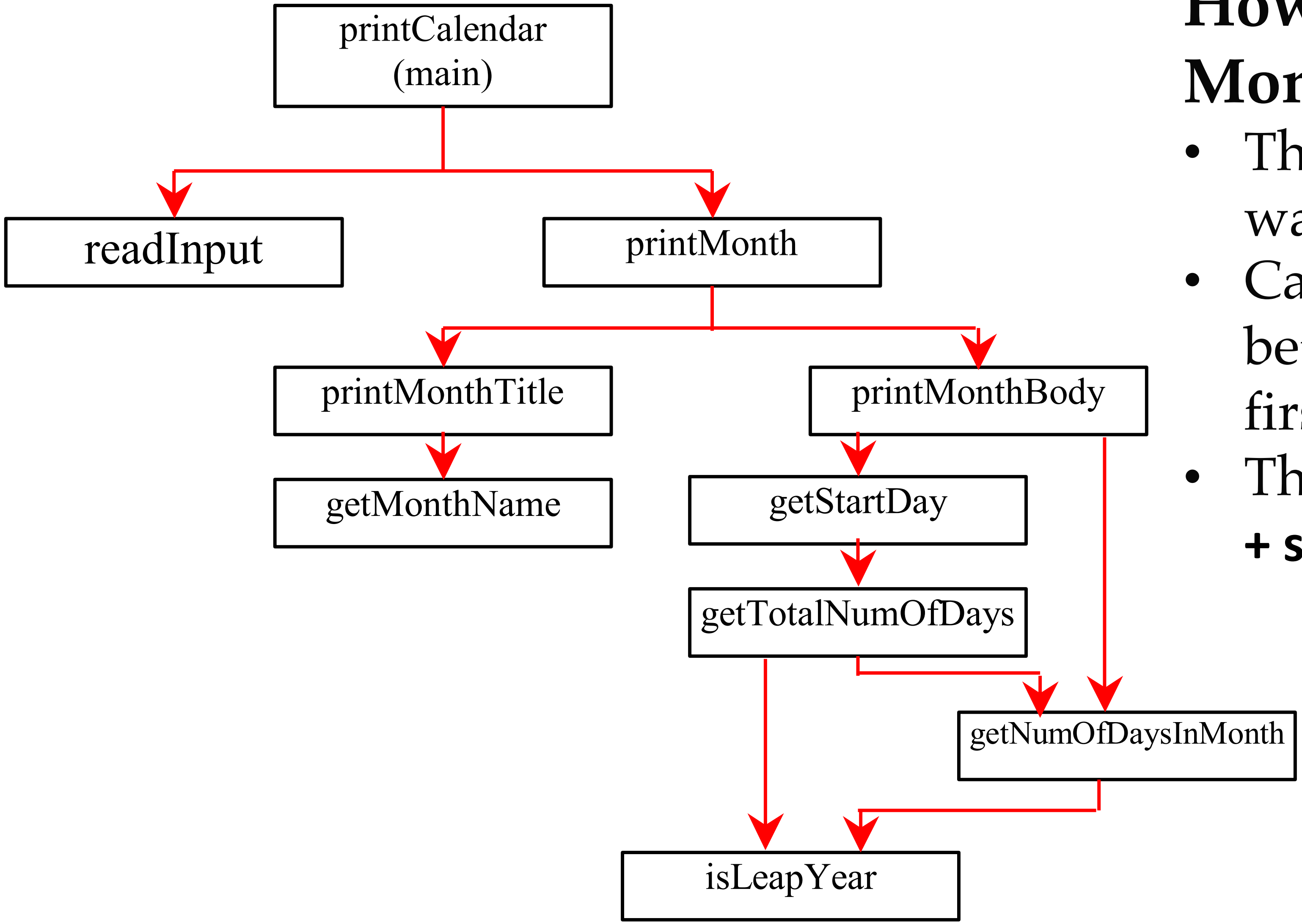
getNumOfDaysInMonth

isLeapYear



```
Enter full year (e.g., 2001): 2014
Enter month as number between 1 and 12: 12
        December 2014
-----------------------------
 Sun Mon Tue Wed Thu Fri Sat
       1   2   3   4   5   6
   7   8   9  10  11  12  13
  14  15  16  17  18  19  20
  21  22  23  24  25  26  27
  28  29  30  31
```

System Design Diagram

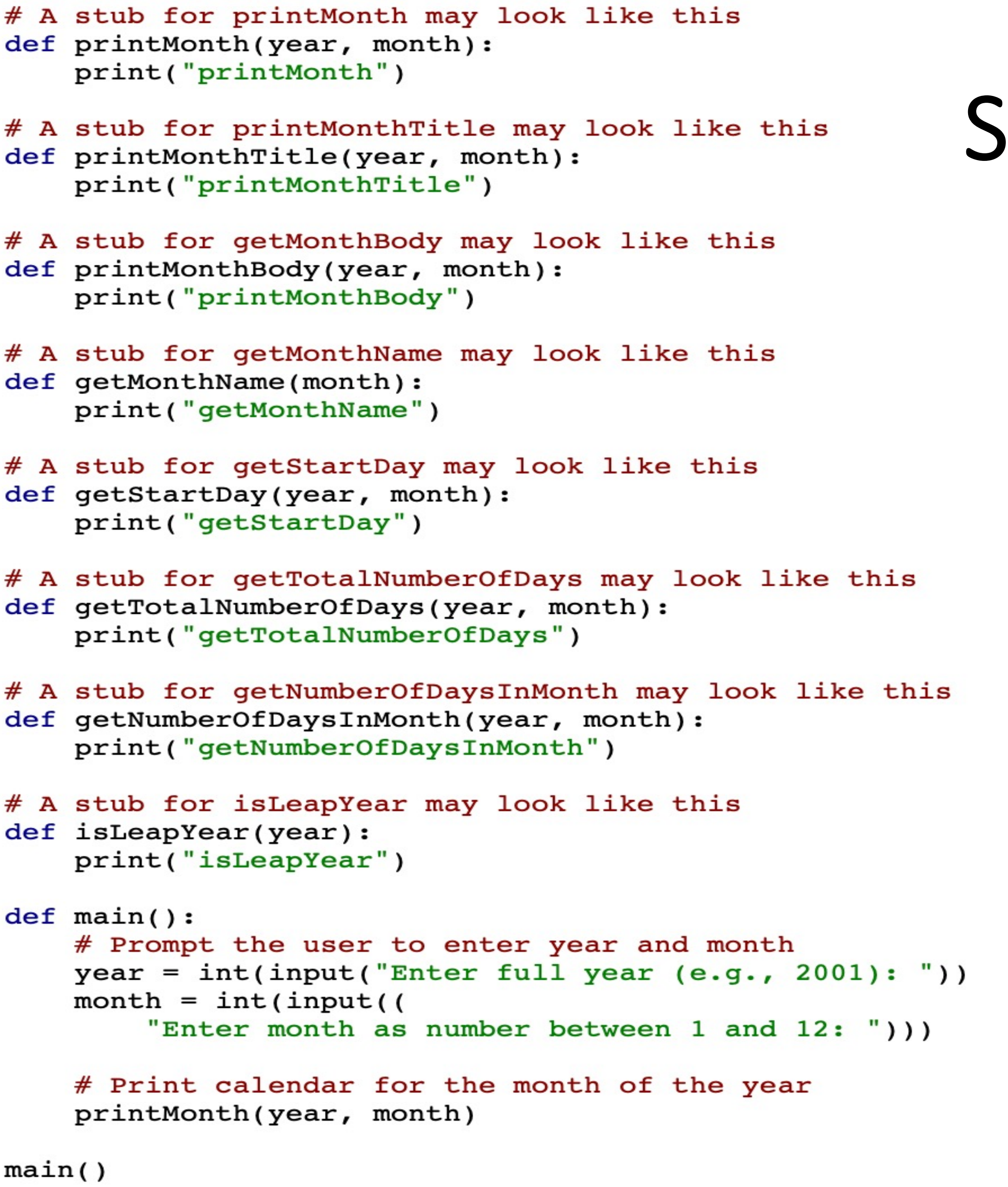


How to get the start day of Month?

- The start day for January 1, 1800, was Wednesday.
- Calculate the total number of days between January 1, 1800, and the first date of the calendar month.
- The start day is **(totalNumberOfDays + startDay1800) % 7**

System Design Diagram

System Design Framework



```
# A stub for printMonth may look like this
def printMonth(year, month):
    print("printMonth")

# A stub for printMonthTitle may look like this
def printMonthTitle(year, month):
    print("printMonthTitle")

# A stub for getMonthBody may look like this
def printMonthBody(year, month):
    print("printMonthBody")

# A stub for getMonthName may look like this
def getMonthName(month):
    print("getMonthName")

# A stub for getStartDay may look like this
def getStartDay(year, month):
    print("getStartDay")

# A stub for getTotalNumberOfDays may look like this
def getTotalNumberOfDays(year, month):
    print("getTotalNumberOfDays")

# A stub for getNumberOfDaysInMonth may look like this
def getNumberOfDaysInMonth(year, month):
    print("getNumberOfDaysInMonth")

# A stub for isLeapYear may look like this
def isLeapYear(year):
    print("isLeapYear")

def main():
    # Prompt the user to enter year and month
    year = int(input("Enter full year (e.g., 2001): "))
    month = int(input((
        "Enter month as number between 1 and 12: ")))

    # Print calendar for the month of the year
    printMonth(year, month)

main()
```

- We use a stub to design the framework of our program/project.

- A **stub** is a simple but incomplete version of a function.

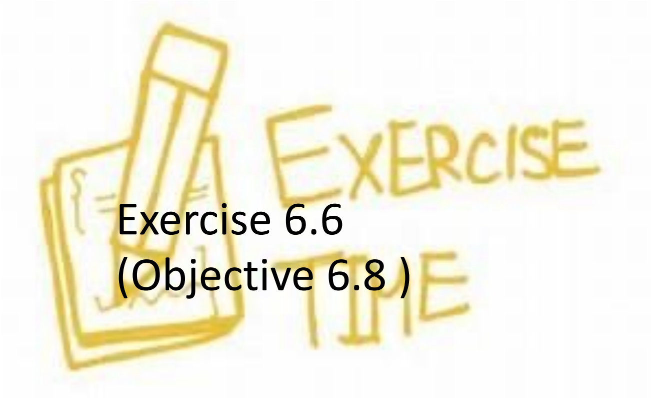
- Enables you to test the framework.

Implementation

- Code available at https://liangcpp.pearsoncmg.com/pyhtml/PrintCalendar.html

- 116 lines

- 9 functions defined



Exercise 6.6
(Objective 6.8)