



ECE12: Introduction to Programming

Lecture 5

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine

Lecture 5: Overview

- Programming principles
 - Algorithm
 - Pseudo code
 - Control flow
- Python keywords
- Control structures
 - Flow charts
 - Selection structures
 - `if` statement
 - `if – else` statement
 - `if – elif - else` statement
- Block indentation
 - `pass` statement
- Structured programming

Programming Principles

- Thorough understanding of the problem
- Problem Definition
 - Input data
 - Output data
- *Algorithm*: Procedure to solve the problem
 - Detailed set of *actions* to perform
 - Specification of *order* in which to perform the actions
 - Termination after a *finite* number of steps
- Pseudo code: Planning a program
 - Informal (English) description of steps in an algorithm
 - Example: Cake baking recipe
- Program: Instructions for the computer
 - Formal description in programming language
 - Statements (steps, actions)
 - Control structures (flow of control)
- Control flow
 - Execution order of statements in the program

Python Keywords

- Keywords in Python

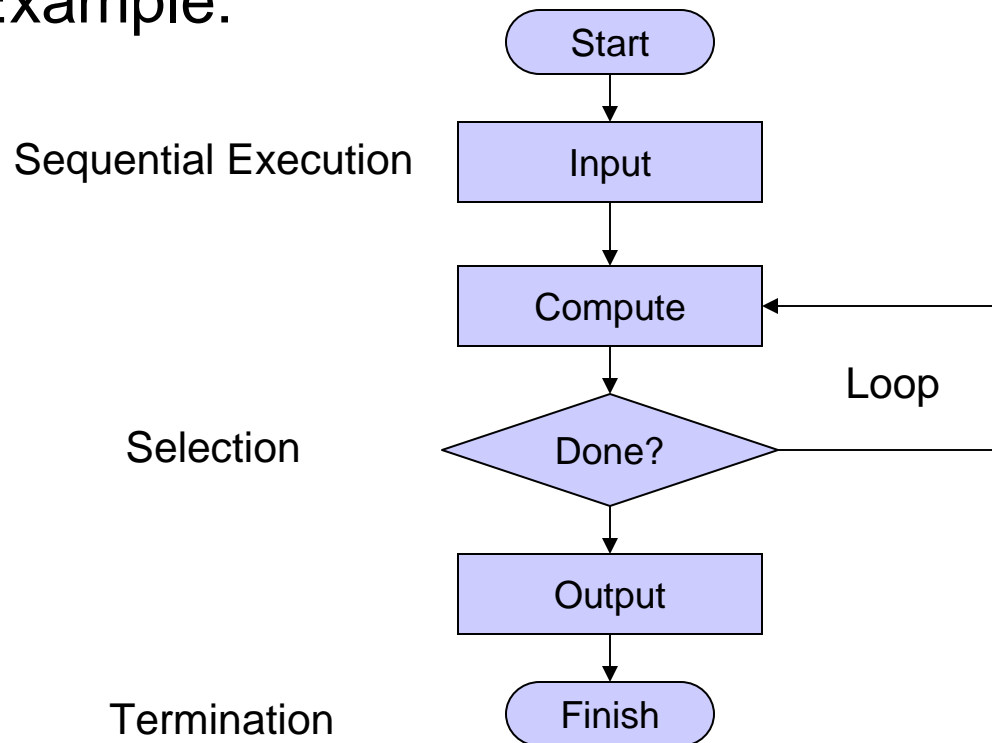
- `and`
- `assert`
- `break`
- `class`
- `continue`
- `def`
- `del`
- `elif`
- `else`
- `except`
- `exec`
- `finally`
- `for`
- `from`
- `global`
- `if`
- `import`
- `in`
- `is`
- `lambda`
- `not`
- `or`
- `pass`
- `print`
- `raise`
- `return`
- `try`
- `while`

- These keywords are reserved and cannot be used as identifiers!
- More keywords may be used in future versions of the language

Control Structures

- Flow Charts

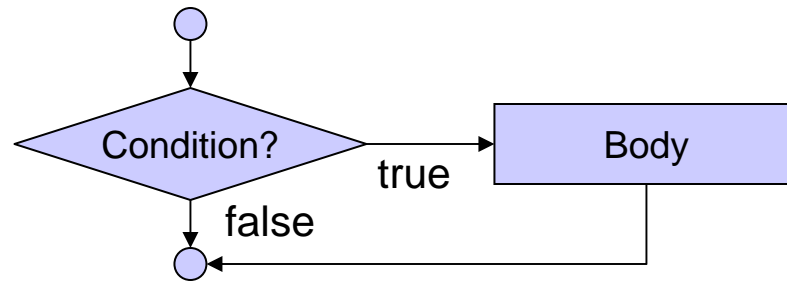
- Graphical representation of program control flow
- Example:



Selection Structures

- **if** statement

- Flow chart:



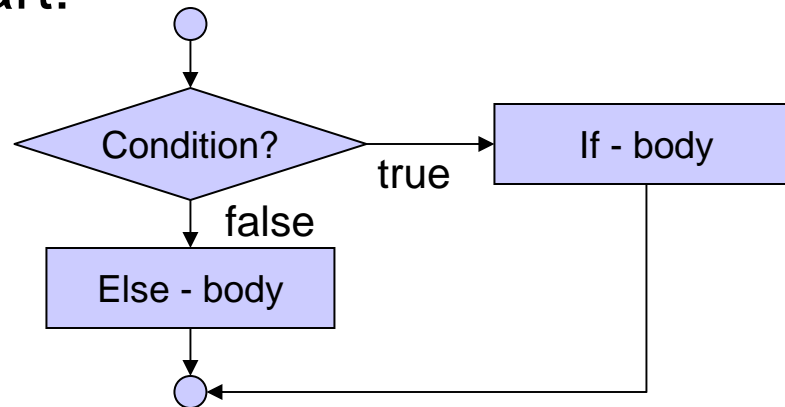
- Example:

```
if grade >= 60:  
    print "Passed."
```

Selection Structures

- **if – else** statement

- Flow chart:



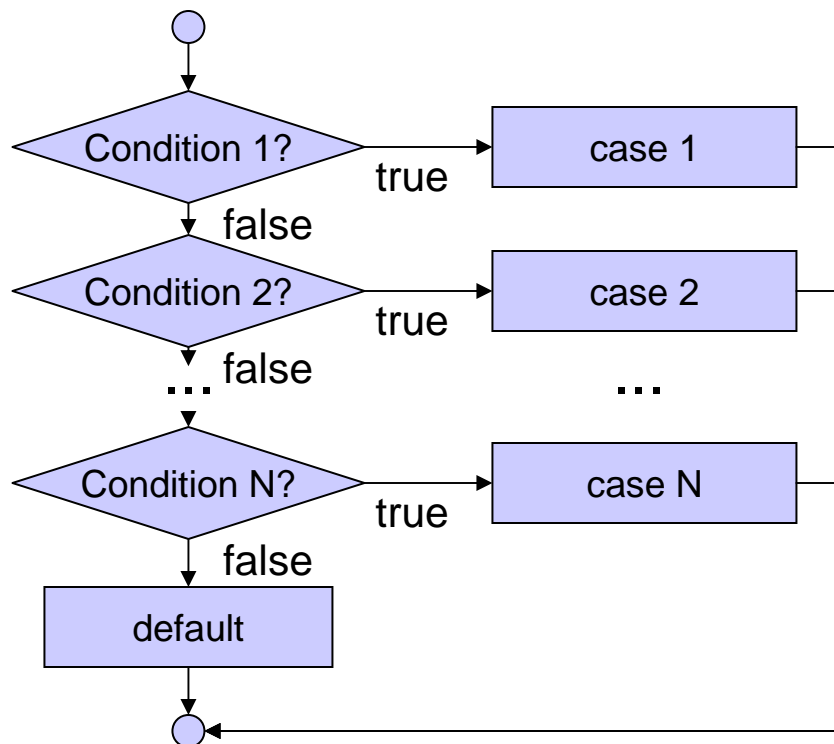
- Example:

```
if grade >= 60:  
    print "Passed."  
else:  
    print "Failed."
```

Selection Structures

- **if – elif – else** statement

– Flow chart:



Example:

```
if grade >= 90:
    print "A"
elif grade >= 80:
    print "B"
elif grade >= 70:
    print "C"
elif grade >= 60:
    print "D"
else:
    print "F"
```


Block Indentation

- Python groups statements into blocks by use of indentation
 - Other languages typically use
 - parentheses `()` e.g. Lisp
 - braces `{ }` e.g. C, C++, Java
 - keywords `begin end` e.g. Pascal
- Example:

```
# some statements...
if x < 0:
    print x, "is negative!"
    # handle negative values of x...
    if x < 100:
        print x, "is too small!"
        # handle the problem
if x > 0:
    # handle positive values of x...
# more statements...
```

- Indentation increases readability of the code
 - in Python, proper indentation is required
 - in other languages, proper indentation is recommended

Block Indentation

- Python groups statements into blocks by use of indentation
 - Other languages typically use
 - parentheses `()` e.g. Lisp
 - braces `{ }` e.g. C, C++, Java
 - keywords `begin end` e.g. Pascal

- Example:

indentation level 0

```
# some statements...
```

```
if x < 0:
```

indentation level 1

```
    print x, "is negative!"
```

```
    # handle negative values of x...
```

```
        if x < 100:
```

indentation level 2

```
            print x, "is too small!"
```

```
            # handle the problem
```

indentation level 0

```
if x > 0:
```

indentation level 1

```
    # handle positive values of x...
```

indentation level 0

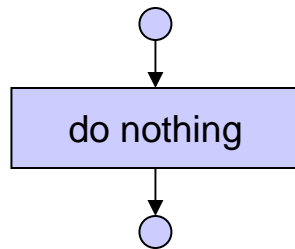
```
# more statements...
```

- Indentation increases readability of the code
 - in Python, proper indentation is required
 - in other languages, proper indentation is recommended

Empty Blocks

- **pass** statement
 - does nothing (no operation, no-op)
 - can be used to represent an empty block
 - Flow chart

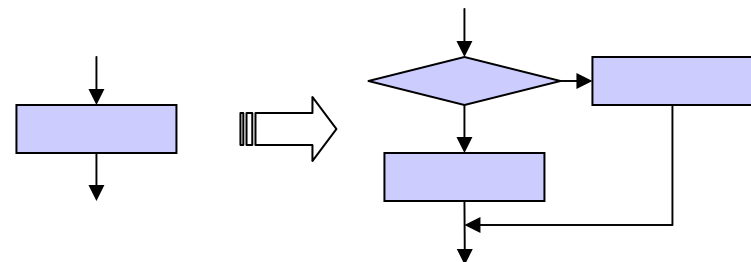
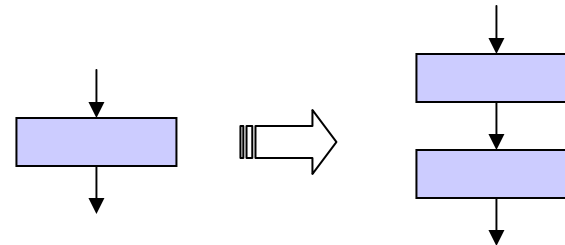
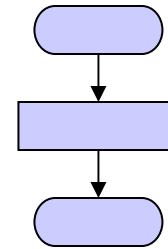
Example:



```
if grade >= 90:
    print "Ask for raise!"
elif grade >= 60:
    pass
else:
    print "Take class again!"
```

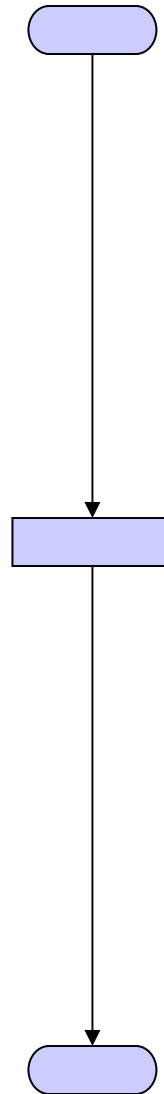
Structured Programming

- Initial flow chart
 - Start
 - Program body
 - Finish
- Statement sequences
 - Statement blocks can be concatenated
 - Sequential execution
- Nested control structures
 - control structures can be placed wherever statement blocks can be placed in the code



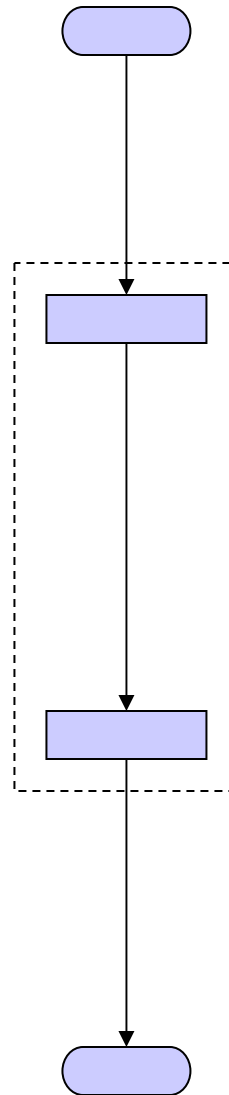
Structured Programming

- Example:
 - Initial flow chart



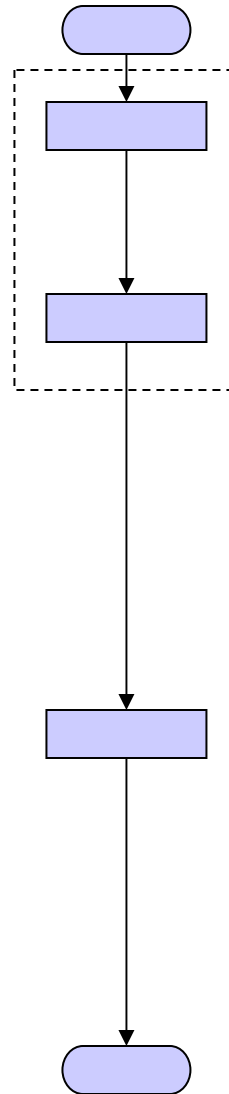
Structured Programming

- Example:
 - Sequential composition



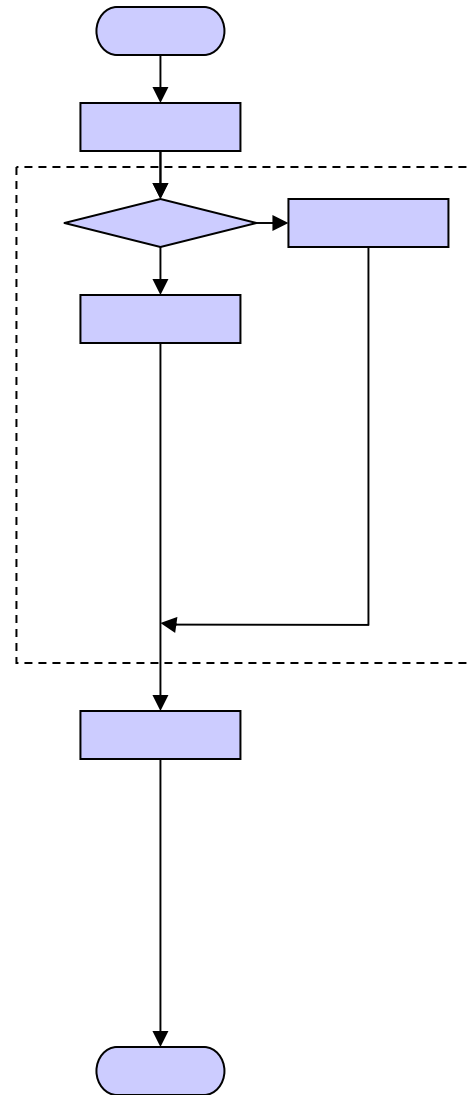
Structured Programming

- Example:
 - insertion of another sequential statement



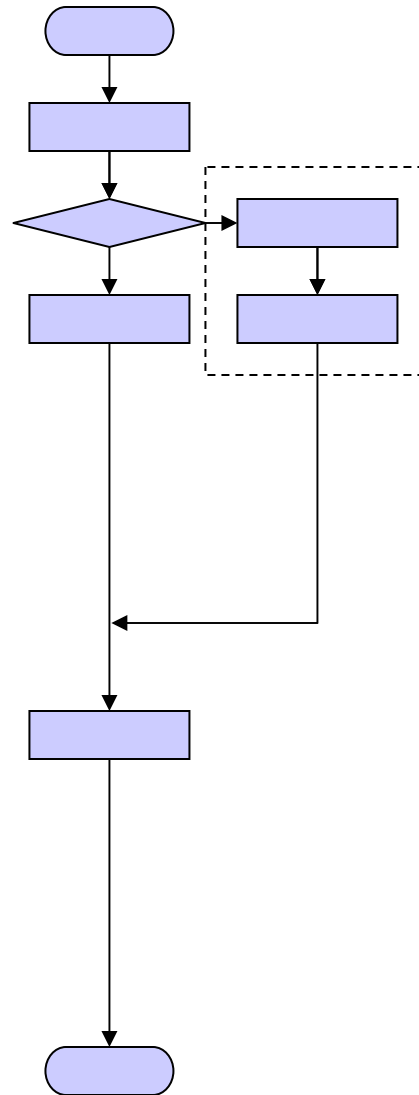
Structured Programming

- Example:
 - insertion of **if – else** statement



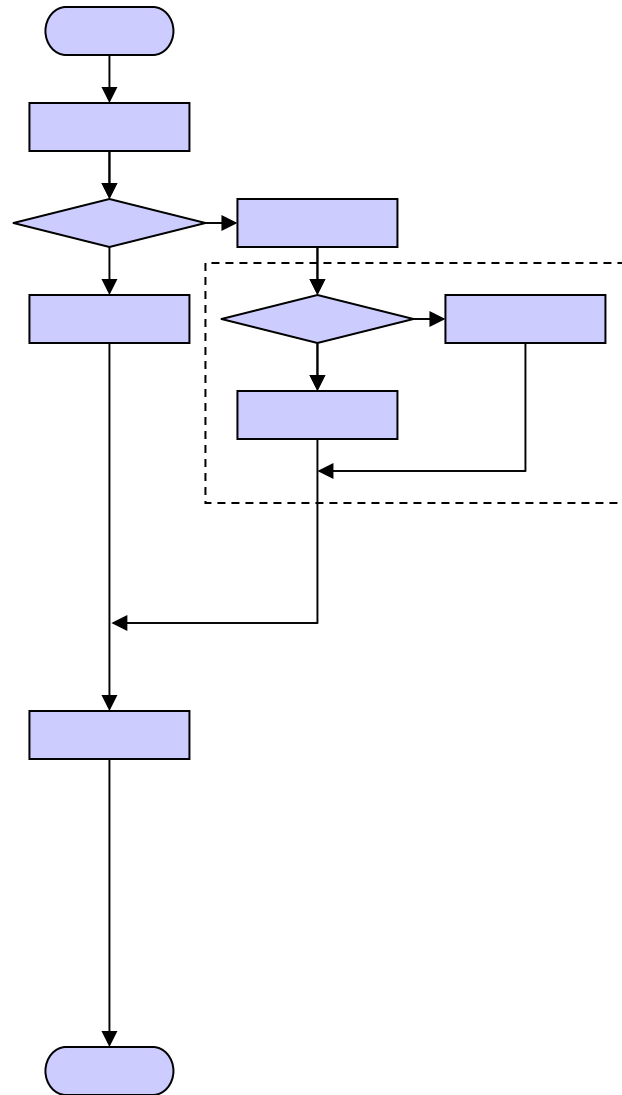
Structured Programming

- Example:
 - insertion of sequential statement



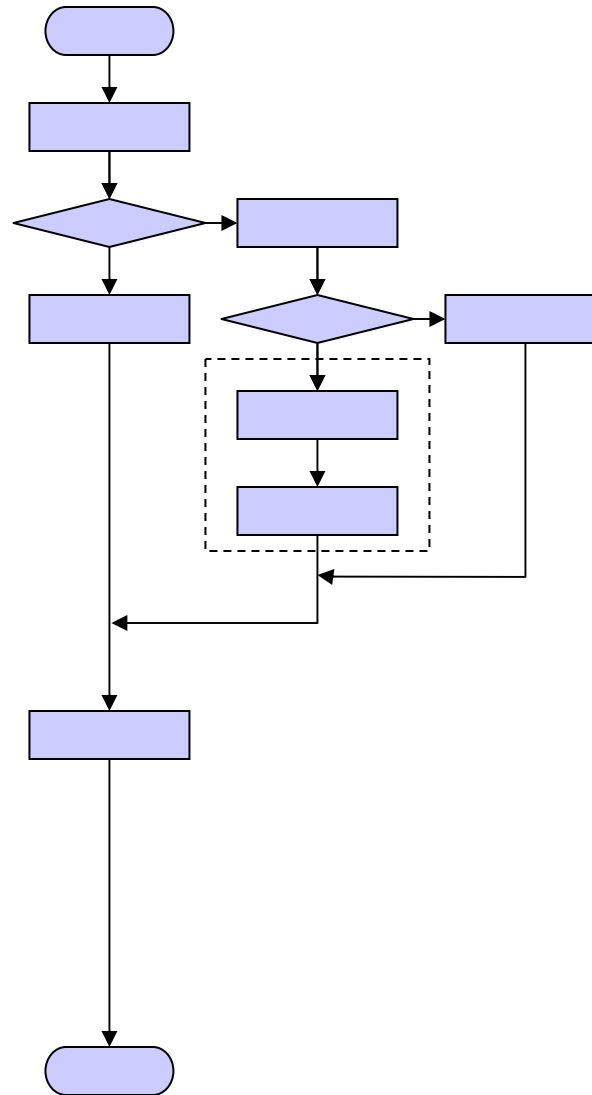
Structured Programming

- Example:
 - insertion of **if – else** statement



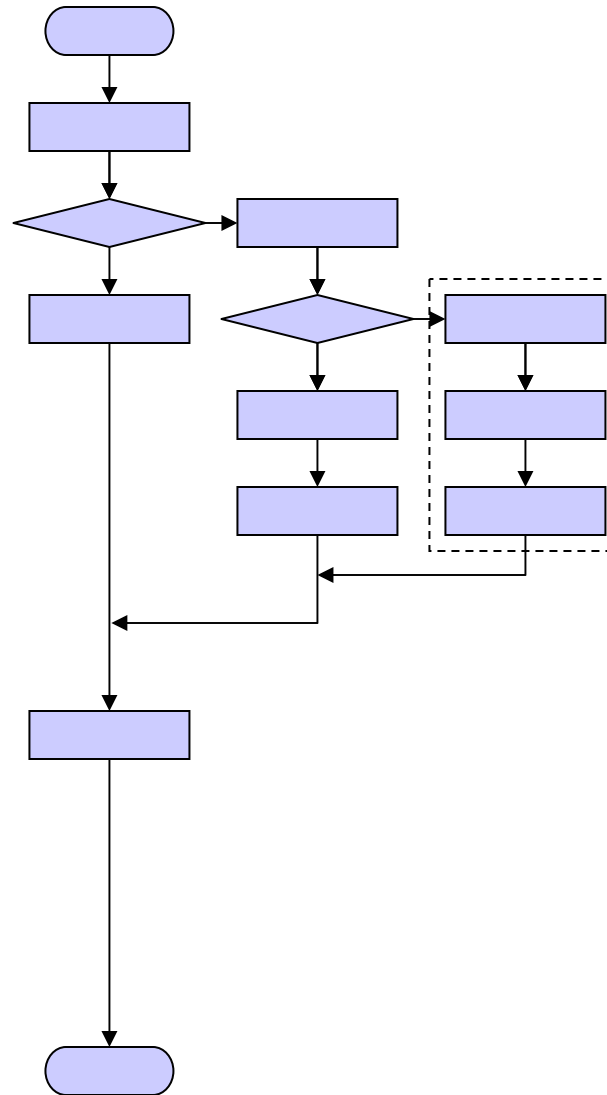
Structured Programming

- Example:
 - insertion of sequential statement



Structured Programming

- Example:
 - insertion of sequential statement (twice)



Structured Programming

- Example:
 - insertion of **if – elif – elif** statement
 - etc. ...

