CSEN 102
Introduction to Computer Science

Lecture 3:

Algorithmic Problem Solving
Conditional Operations

Prof. Dr. Slim Abdennadher
Dr. Nada Sharaf
Dr. Mohammed salem, slim.abdennadher@guc.edu.eg,
nada.hamed@guc.edu.eg,
mohammed.salem@guc.edu.eg

German University Cairo, Department of Media Engineering and Technology

31.10.2020 - 05.11.2020

1 Synopsis

1.1 Sequential operations

Synopsis

- What is computer science?
- What is an algorithm?

  Definition 1 (Algorithm)**.** An algorithm is a *well-ordered* collection of *unambiguous* and *effectively computable operations* that, when executed, *produces a result* and *halts in a finite amount of time*.

- Why python?
- What are the necessary elements for sequential algorithms?
  - *Input* (*e. g.*, “`A = eval(input())`”)
  - *Output* (*e. g.*, “`print(A)`”, or “`print("text")`”)
  - *Calculation*, *manipulation* (*e. g.*, “`A = B + C`”)

Sequential operations

Example 2 (See last lecture)*.* For a given number of eggs, find out how many dozen eggs we have and how many extra eggs are left over.

```
eggs = eval(input())
dozens = int(eggs / 12)
extras = eggs - (dozens * 12)
print("Your␣number␣of␣eggs␣is␣")
print(dozens)
print("␣dozen(s)␣and␣")
print(extras)
print("␣extra(s)")
```

- Let the input be 27
- What is the output?

Where the function `int` rounds down the result to an integer. For example `int(10/3) = 3`.

How to construct an algorithm

- Identify the *input* of the algorithm
- Introduce *variables* for
  - *Input*
  - (intermediate) *results*
- Analyze the task into *sequential steps*
- Provide for detailed *output*

2 Conditional operations

2.1 Introduction

Objectives

By the end of this lecture, you should be able to:

- *Design algorithms using conditional operations*

Algorithms: operations

Algorithms can be constructed by the following operations:

- Sequential Operation
- *Conditional Operation*
- *Iterative Operation*

2.2 Concepts

Conditional operation – idea

Decision

```
nameOfSinger = input()
if nameOfSinger == ’Mohamed Mounir’:
    print(’I will go home’)
else:
    print(’I will stay at the GUC’)
```

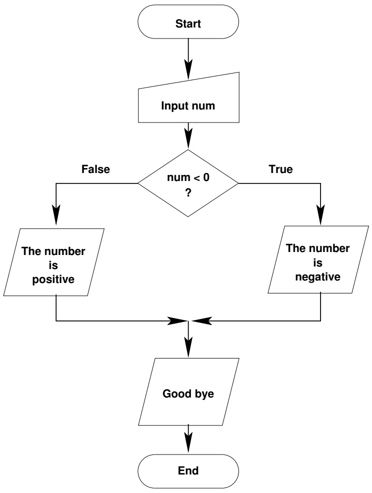
Conditional operation – principle

- Rationale
  - Determines whether or not a condition is true; and based on whether or not it is true; *selects the next step* to do
- Notation
  - Use the same primitives as before plus the following:

```
if condition:
        # <operations for the then-part>
else
        # <operations for the else-part>
```

- Execution
  - Evaluate `<condition>` expression to see whether it is true or false.
  - If true, then execute operations in `if`-part
  - Otherwise, execute operations in `else`-part

Conditional operation – diagram



```
num = eval(input())
if num < 0:
    print("The number is negative")
else:
    print("The number is positive")

print("Good-bye for now")
```

2.3 General examples

Conditional operation – examples

Example 1:
Write an algorithm to compute the absolute value of a given number.

```
Number = int(input())

if (Number >= 0):
        Value = Number
else:
        Value = (-1) * Number

print(Value)
```

Conditional operation – examples

Example 2:
Give the user a choice of seeing the area or the circumference of a circle given its radius.

```
radius = eval(input())
response = input("Type␣A␣for␣area␣or␣C␣for␣circumference")
if (response == "A"):
 area = (radius * radius * 3.14)
 print(area)
else:
 circumference = (2 * radius * 3.14)
 print(circumference)
```

Conditional Operation – examples

Example 3:
Write an algorithm to convert Euro (EUR, €) to Egyptian Pound (EGP, £E) and Egyptian Pound to Euro. The inputs to your algorithm are the following:

- Amount of money to be converted
- Conversion Type (*i. e.*, 1 for EUR to EGP and 2 for EGP to EUR)
- Exchange Rate (*i. e.*, the EGP equivalent for 1 EUR)

```
amount, type, rate = eval(input()), eval(input()), eval(input())
if type == 1:
    amount = amount * rate
else:
    amount = amount / rate
print(amount)
```

2.4 Compounded conditions

Compounded conditions

Conditions may be *compounded* using AND, OR and NOT.

- E1 or E2: true if at least one of them is true; false otherwise.
- E1 and E2: true if both are true; false otherwise.
- not E: true if E is false and false if E is true.

Find the sum of three positive numbers

```
A, B, C = int(input()), int(input()), int(input())
if (A > 0) and (B > 0) and (C > 0):
 Sum = (A+B+C)
 print(Sum)
```

2.5 Nested conditions

Conditional algorithms with more than two choices

Nested if-statement

```
if first_condition:
    # <do first thing>
else:
    if second_condition:
        # <do second thing>
    else:
        # <do something else>
```

2.6 Examples for nested conditions

Nested if-statement – examples

Example 1
Algorithm to find the largest of three numbers.

```
A, B, C = eval(input()), eval(input()), eval(input())
if A >= B:
    if A >= C:
        print(A)
    else:
        print(C)
else:
    if B >= C:
        print(B)
    else:
        print(C)
```

Nested if-statement – examples

Example 2
Write an algorithm that reads each student's marks, print either a grade or an error message. Students marks in a class are graded on the following policy:

- A: 85-100
- B: 74-85
- C: 60-74
- D: 50-60
- F: <50

Nested if-statement – examples

Solution with if and else

```
Mark = eval(input())
if(Mark >=0):
    if (Mark >100):
        print("invalid␣mark")
    else:
        if (Mark <50):
            print("grade is F")
        else:
            if (Mark <60):
                print("grade is D")
            else:
                if (Mark <74):
                    print("grade is C")
                else:
                    if (Mark <85):
                        print("grade is B")
                    else:
                        print("grade is A")
```

Nested if-statement – examples

Solution with if and elif

```
Mark = eval(input())

if(Mark >=0):
    if (Mark >100):
        grade = "invalid␣mark"
    elif (Mark <50):
        grade = "grade is F"
    elif (Mark <60):
        grade = "grade is D"
    elif (Mark <74):
        grade = "grade is C"
    elif (Mark <85):
        grade = "grade is B"
    else:
      grade = "grade is A"

print(grade)
```

Nested if-statement – examples

Example 3
Given an employee's eligible medical expenses for a calendar year, write an algorithm which computes the amount of reimbursement from group medical insurance.

- The insurance does not cover the first 100 LE of medical expenses.
- It pays 90% of the remaining amount in the first 2000 LE of expenses and 100% of any additional expenses.

Nested if-statement – examples

```
LL = 100
UL = 2000
Expense = int(input())

if (Expense < LL):
        Refund = 0
else:
        if (Expense < UL):
                Refund = 0.9 * (Expense-LL)
        else:
                Refund = 0.90 * (UL-LL) + (Expense - UL)

print(Refund)
```