*UCT Department of Computer Science*
*Computer Science 1015F*

# Selection


*Hussein Suleman*
*<hussein@cs.uct.ac.za>*
*Brian DeRenzi*
*<bderenzi@cs.uct.ac.za>*

*February 2016*

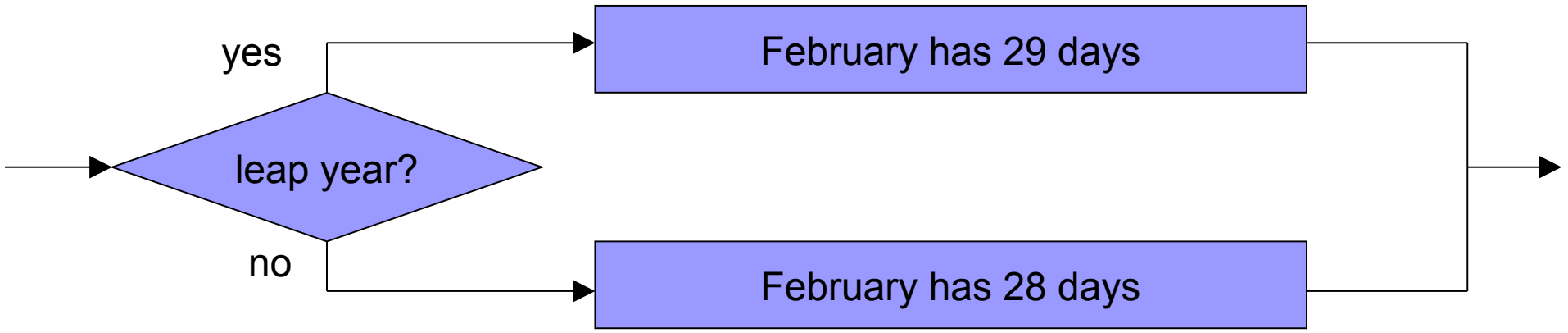
# Problem 1 Introduction

- Write a program to calculate the minimum of 4 integers without using the min/max/math functions. Use a sequence of *if* statements.



# What is Selection?

- Making choices in the flow of execution of a program.
  - e.g., if it is a leap year then there are 29 days in February – otherwise there are 28



# Conditional expressions

- Selections are made on the basis of expressions that must evaluate to True or False (boolean).
- Relational operators always return boolean values, e.g.:
  - `answer > 1.0`
  - `numberOfPeople <= 14`
  - `month == 12   // note: not the same as "="`
  - `date != 13       // not equal`
  - `money >= 5000`



# The “if” statement

```
if boolean_expression:
    statement1
    statement2
    …
else:
    statementa
    statementb
    …
```

- Statements must be indented to same level to be considered part of the same block.
- Python will usually execute all statements within a block once it start on a block.



# Example usage

```
if month == 12:
    print ("Hoorah! No classes")
else:
    print (":-(")
```



# Another example

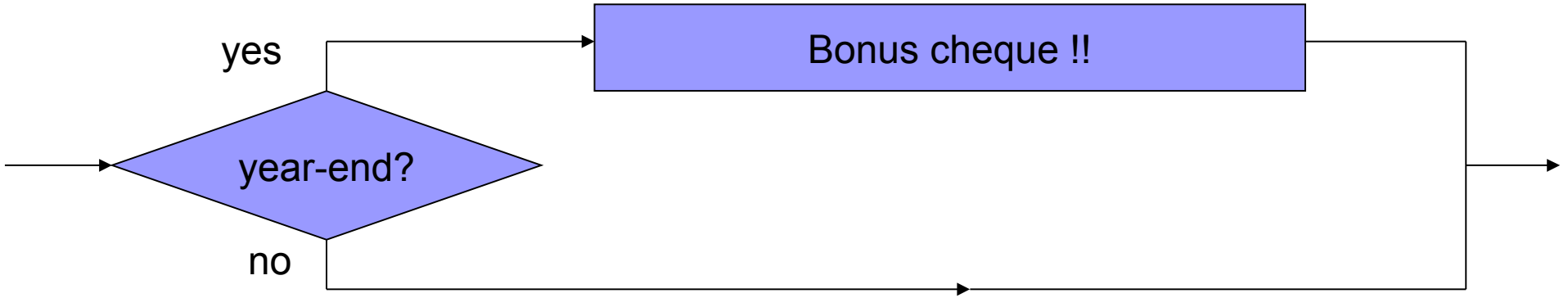
```
if year < 2000:
   fearFactor = 1
else:
   fearFactor = 0

if fearFactor == 1:
   print (“be afraid – be very afraid”)
else:
   print (“it’s OK! no Y2K bug!”)
```



# Shortcut

- No else part.

```
if numberOfStudents > 150:
    print ("Full!")
```



# Problem 1

- Write a program to calculate the minimum of 4 integers without using the min/max/math functions. Use a sequence of *if* statements.



# Problem 2 Introduction

- Write a program to calculate the minimum of 4 integers without using the min/max/math functions. Use nested *if* statements.



# Nested “if” statement

```
password = input ("Enter your password")
if password==realPassword:
    if name==“admin”:
        loggedIn = true
else:
    print ("Error")
```



# Dangling Else

- Python can always determine which “if” an “else” belongs to because they have the same indent level.

```
password = input ("Enter password:")
if password==realPassword:
    if name==“admin”:
        loggedIn = true
    else:
        print (“Error”)
```

- In other programming languages, this is called the **dangling else** problem.  Python does not have this problem.

# Multiway selection

- Multiple conditions, each of which causes a different block of statements to execute.
- Can be used where there are more than 2 options.

```
if condition1:
    statements …
else:
    if condition2:
        statements …
    else:
        if condition3:
            statements …
        else:
        …
```



# “if” ladder

- Just a nicer way to write multiway selection.

```
if operation == ‘a’:
   answer = first + second
elif operation == ‘s’:
   answer = first - second
elif operation == ‘m’:
   answer = first * second
```



# Problem 2

- Write a program to calculate the minimum of 4 integers without using the Math methods. Use nested *if* statements.



# Problem 3

- Write a program to sort 3 integers and output the sorted order symbolically. For example, if the numbers are {a=3, b=6, c=5}, then the sorted order is “a c b”.
- Use nested *if* statements.



# Problem 4

- Write a program to calculate your final grade and symbol in CSC1015F based on marks for theory tests, exam, practicals and practical tests.  This must include the possibility of DPR.



# Problem 5 Introduction

- Write a program to calculate the minimum of 4 integers without using the min/max/math functions. Use *if* statements with boolean expressions.



# Booleans Revisitied

- boolean – stores only *True* or *False* values.
  - e.g., `iLikeCSC1015 = True`

```
if iLikeCSC1015:
    iEatWeetbix = True
```



# Boolean operators

| Boolean Algebra | Python | Meaning |
|---|---|---|
| AND | and | true if both operands are true |
| OR | or | true if at least one operand is true |
| NOT | not | true if operand is false;<br>false if operand is true |



# Operator precedence

- Now that we have seen how operators can be mixed, we need precedence rules for all operators
  - () (highest precedence – performed first)
  - **
  - * / // %
  - + -
  - < <= >= > == !=
  - not
  - and
  - or (lowest precedence – performed last)



# Reversing expressions

- Use **not** operator to reverse meaning of boolean expression, e.g.,

```
if mark >= 0:
# do nothing
else:
   print ("Error")
```

- Instead, invert the condition

```
if not (mark >= 0):
   print ("Error")
```

- Can we do better ?



# Boolean operator example

```
inClassroom = False
isRaining = True
…
if (inClassroom and isRaining):
   print (“Lucky!”)
…
if (not inClassroom and isRaining):
   print (“Wet and miserable!”)
…
if (not isRaining and not inClassroom):
   print (“Happy!”)
```

- Homework: What are De Morgan's Laws



# Boolean expression example

```
marks = ...
…
if marks >= 75:
    symbol = 'A'
…
if (marks >= 65 and marks < 75):
    symbol = 'B'
…
if (marks < 0 or marks > 100):
    symbol = 'X'
    print ("Invalid mark!")
```



# Problem 5

- Write a program to calculate the minimum of 4 integers without using the min/max/math functions. Use *if* statements with boolean expressions.



# Problem 6

- Write a program to check the login name and password for an online system such as Vula. Your program must assume a set of 3 valid users and check only for those users, outputting an appropriate message in either case.



# Problem 7

- Write a program to determine the ingredients in a sandwich based on the sandwich number.



# Problem 8 Introduction

- Write a program to perform a selectable standard operation (+-/*) on a pair of numbers depending on an operation specified as an input value of either 'a', 'm', 's' or 'd'.
- For example, if the numbers are entered as 3 and 5 and the operation is entered as 'm', the result should be 15.



# Problem 8

```
a = eval (input ("Enter number a: "))
b = eval (input ("Enter number b: "))
operation = input ("Enter operation [a/m/d/s]: ")

if operation=="a":
   answer = a+b
elif operation=="s":
   answer = a-b
elif operation=="m":
   answer = a*b
elif operation=="d":
   if b!=0:
      answer = a/b
   else:
      answer = 0
      print ("Error")
else:
   answer = 0
   print ("Error")

print (answer)
```