SCC461 – Programming for Data Scientists

Leandro Marcolino

Lancaster University

Week 6

Outline

Overview

- Weeks 1 to 5:
  - Taught by Tom
  - Focused on the statistical side of programming using R
- Weeks 6 to 10
  - Taught by Leandro
  - Object-oriented programming and Python

Tentative Schedule

- Week 6: Basics of programming
- Week 7: Basics + Object Oriented programming I
- Week 8: Object Oriented Programming II + Libraries
- Week 9: Libraries and Problem Solving I
- Week 10: Libraries and Problem Solving II

Assessment

- Weekly Assessment (25%)
  - Deadline: 9am the following Monday
  - Source Code
  - Solutions to test cases
  - Short text on your learning/problem solving experience
  - Short reply about a paper
  - More details later today...
- Assignment (50%)
  - Details will be published in week 8, due in week 11 (after Chirstmas)

Teaching Assistants

Elnaz
e.shafipouryourdshahi@lancaster.ac.uk

Yao
y.zhang70@lancaster.ac.uk

Outline

Programming?

- Step by step instructions
- Instructions have an effect

Why should I care?

Create cell phone apps

Create dynamic web pages

Research

Data integration, collection, analysis

Programming Languages

0	0	0	1	0	1	0	1	0	1	0	1	0	1	1	1	1	0
1	0	0	1	1	0	1	0	1	0	1	0	0	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	1	1	1	1	1	0	0	0	0	1
0	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0	1	1	1	1	1	0	0	0
1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	1	0	1	0	1	0	1	1	1	1	1	1	1	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1
1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
0	0	0	0	0	0	1	0	1	0	1	1	1	1	0	1	0

Repeat 10 times:

Print "Programming is Fun!"

Compiler vs Interpreter

```
Repeat 10 times:
        Print "Programming is Fun!"
```

```
Repeat 10 times:
        Print "Programming is Fun!"
```

```
0 0 0 1 0 1 0 1 0 1 0 1 0 1 1 1 1
0 1 0 0 1 1 0 1 0 1 0 1 0 0 0 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1
1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 1 1 1 1 1 0 0 0 0
1 0 1
```

Compiler

```
Programming is Fun!
Programming is Fun!
Programming is Fun!
Programming is Fun!
Programming is Fun!
Programming is Fun!
Programming is Fun!
Programming is Fun!
Programming is Fun!
Programming is Fun!
```

Interpreter

Python

"Python is powerful... and fast; plays well with others; runs everywhere; is friendly & easy to learn; is Open."

Python

In this course we will study Python as an **Example**! But...

NEVER MARRY A PROGRAMMING LANGUAGE!

Python

- This course...
  - ... is about learning how to program
  - ... is about sharpening your programming skills

Python

The language reference is your friend!

Python

References

- https://docs.python.org/3/
- http://openbookproject.net/thinkcs/python/english3e/

Integrated Development Environment

eric5 - /home/detlev/Development/Python/Eric5/eric5_5/eric5/Plugins/PluginSyntaxChecker.py - eric5

File Edit View Start Debug Unittest Multiproject Project Refactoring Extras Settings Window Bookmarks Plugins Help

eric5.py | ...csPlugins/vcsMercurial/HgStatusDialog.py

HgStatusDialog | __commit

```
588         """
589         self.__restoreMissing()
590
591     ##########################################################################
592     ## Context menu handling methods
593     ##########################################################################
594
595     def __showContextMenu(self, coord):
596         """
597         Protected slot to show the context menu of the status list.
598
599         @param coord the position of the mouse pointer (QPoint)
600         """
601         # TODO: set status of menu entries according to their conditions
602         if self.vcs.isExtensionActive("largefiles"):
603             enable = len(self.__getUnversionedItems()) > 0
604         else:
605             enable = False
606         for act in self.lfActions:
607             act.setEnabled(enable)
608         self.menu.popup(self.mapToGlobal(coord))
609
610     def __showAddMenu(self):
611         """
```

Plugins/PluginSyntaxChecker.py

SyntaxCheckerPlugin

```
34
35  error = ""
36
37
38  class SyntaxCheckerPlugin(QObject):
39      """
40      Class implementing the Syntax Checker plugin.
41      """
42      def __init__(self, ui):
43          """
44          Constructor
45
46          @param ui reference to the user interface object (UI.UserInterface)
47          """
48          super().__init__(ui)
49          self.__ui = ui
50          self.__initialize()
51
52      def __initialize(self):
53          """
54          Private slot to (re)initialize the plugin.
55          """
56          self.__projectAct = None
57          self.__projectSyntaxCheckerDialog = None
```

Summary | Filename

TODO: add support for hg summary --mq | Plugins/VcsPlugins/vcsMercurial/QueuesExtensi

TODO: release - reenable redirection | UI/UserInterface.py

Channels

[Arfrever]
Arfrever
ChanServ
detlev

[13:09] [-->] You have joined the channel #eric-ide (~detlev@2a02:3100:1601:3e00:d08:3175:dee2:be10).
[13:09] [***] Channel modes: no colors allowed, no messages from outside, topic protection.
[13:09] [***] This channel was created on 2012-11-15 18:36.
[13:09] [***] Channel URL: http://eric-ide.python-projects.org

Enter a message, send by pressing Return or Enter

#eric-ide (5)

Network

[13:09] [MOTD] - together with like-minded FOSS enthusiasts for talks and
[13:09] [MOTD] - real-life collaboration, if you're more keen on the outdoors why
[13:09] [MOTD] - not attend or arrange a local geeknic (http://www.geeknic.org).
[13:09] [MOTD] -
[13:09] [MOTD] - We would like to thank Private Internet Access
[13:09] [MOTD] - (https://www.privateinternetaccess.com/) and the other
[13:09] [MOTD] - organisations that help keep freenode and our other projects
[13:09] [MOTD] - running for their sustained support.
[13:09] [MOTD] -
[13:09] [MOTD] - In particular we would like to thank the sponsor
[13:09] [MOTD] - of this server, details of which can be found above.
[13:09] [MOTD] -
[13:09] [MOTD] - **********
[13:09] [MOTD] - Please read http://blog.freenode.net/2010/11/be-safe-out-there/
[13:09] [MOTD] - **********
[13:09] [MOTD] End of message of the day

Before we start...

Show Scratch examples

Outline

Fundamentals

- Commands
- Variables
- Lists
- Loops
- Conditionals
- Functions

Print

```
print ("Hello! This is my first command!");
```

Expressions

```
2 + 3 * 5;

11/2;

11%2;

2 ** 3;
```

Variables

$$\boxed{\begin{aligned} x &= 1 \\ x &= x + 1 \end{aligned}}$$

Variables

$$x = 1$$

$$\cancel{x} = \cancel{x} + 1$$

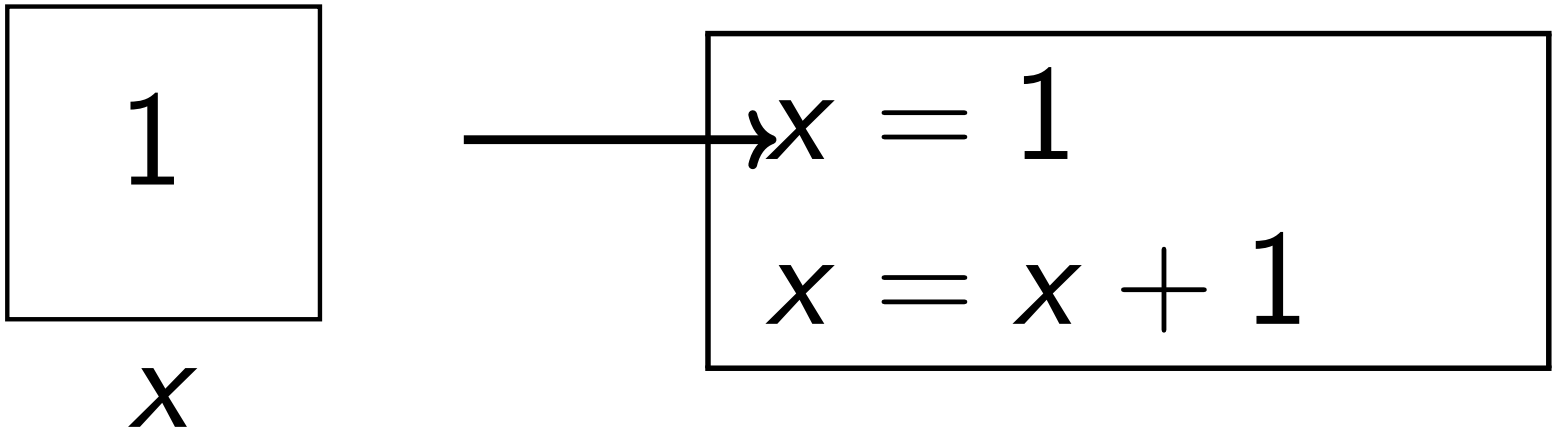
Variables

$$x = 1$$

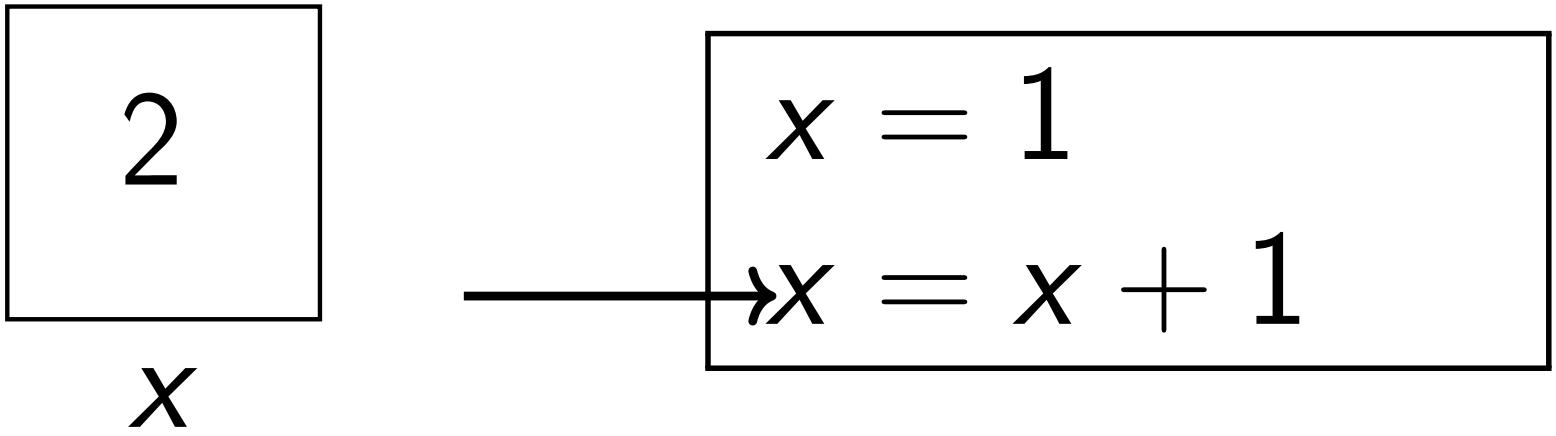
$$\cancel{x} = \cancel{x} + 1$$

$0 = 1???$

Variables



Variables



Variables

```
x = 1;
x = x + 1;
print(x);
```

Variables

```
x = 1;
x = x + 1;
print("The value of x is " + str(x));
```

Variables

```
x = 1;
x = x + 1;
y = x;
print("The value of y is " + str(y));
```

Variables

```
a = 0;
b = a + 5*3;
a = b * 5;
print(a);
```

What will be printed?

Variables

Strings

```
language = "Python";

print("I love " + language);
```

Conditionals

```
a = 0;
b = a + 5*3;
a = b * 5;

if (a > b):
        print("Hey!");
```

Conditionals

```
a = 0;
b = a + 5*3;
a = b * 5;

if (a > b):
        print("Ho!");
```

Conditionals

```
a = 0;
b = a + 5*3;
a = b * 5;

if (a == b):
        print("Let's go!");
```

Conditionals

```
a = 0;
b = a + 5*3;
a = b * 5;

if ((a % b) == 0):
        print("b divides a");
else:
        print("b DOES NOT divide a");
```

User Input

```
n = input("Enter a number, and I will multiply
    it by 2: ");
print(n*2);
```

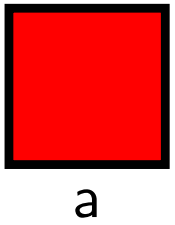
Conditionals

Exercise

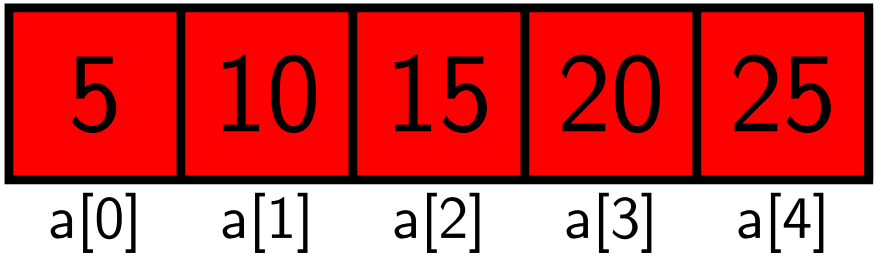
- Ask user to input a number
- Print whether it is divisable by 5

Lists

a = 1;



a = [5, 10, 15, 20, 25];



Lists

```
myList = [5, 10, 15, 20, 25];

print(myList);

print(MyList[0]);
print(MyList[2]);
print(MyList[5]);
```

Lists

```
myList = [5, 10, 15, 20, 25];

print(myList);

print(MyList[0]);
print(MyList[2]);

myList[2] = myList[3] - myList[4];

print(myList[2]);
```

Lists

Slices

```
myList[2:5]

myList[2:];

myList[:2];
```

Lists

Operations

```
a = [1, 2, 3];
b = [4, 5, 6];
c = a + b;
print(c);
```

Lists

Operations

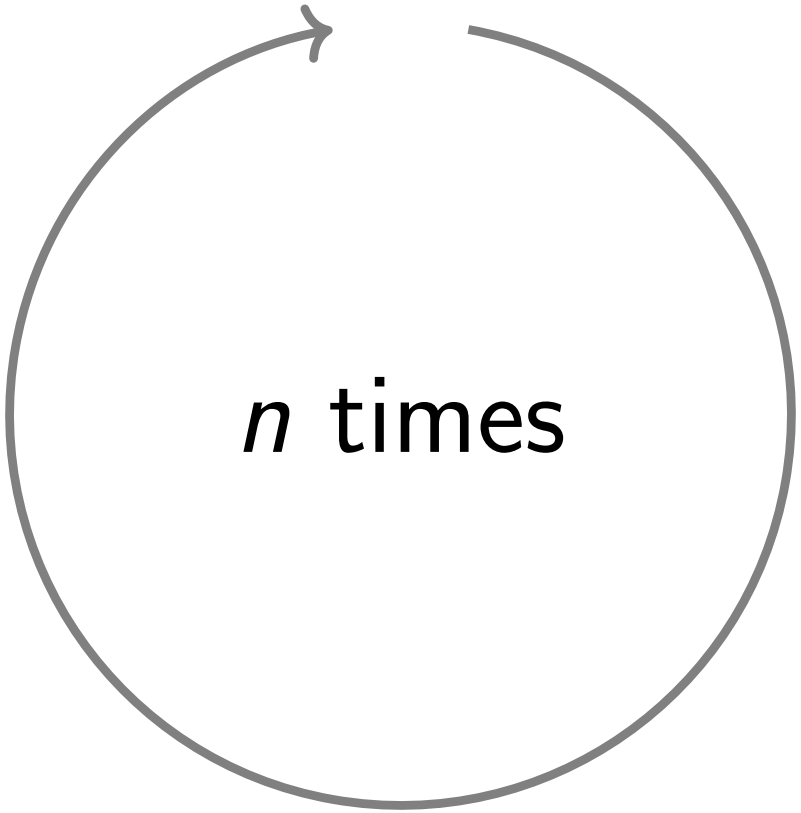
```
a = [0] * 4;
print(a);

b = [1, 2, 3] * 3;
print(b);
```

Lists

Strings

```
languages = ["Python", "Java"];

print("I love two languages: " + languages[0] +
    " and " + languages[1]);
```

Iterations (Loops)



Iterations

```
print(1);
print(2);
print(3);
print(4);
```

Iterations

```
printMe = 1;

while (printMe <= 4):
        print(printMe);
        printMe = printMe + 1;
```

Iterations

```
printMe = 1;

while (printMe <= 9999):
        print(printMe);
        printMe = printMe + 1;
```

Iterations

For loops

```
for f in [1,2,3,4]:
        print(f);
```

Iterations

For loops

```
for f in [1,2,3,4]:
        print(f);
```

What about 9999???

Iterations

For loops

```
for f in range(1,10000):
        print(f);
```

Iterations

Exercise

- Ask user to input a number
- Print whether it is prime

Iterations

Exercise

- Create a list with the first 100 prime numbers
- Print the elements of the list

Functions

$$x = 1;$$
$$x = x + 1;$$
$$\text{print}(x);$$

Functions

$x = 1;$
$x = x + 1;$
$\text{print}(x);$

Function *print(x)*
Initialize screen;
Open buffer;
Write x to buffer;
Close buffer;

Functions

Why do we use functions?

Functions

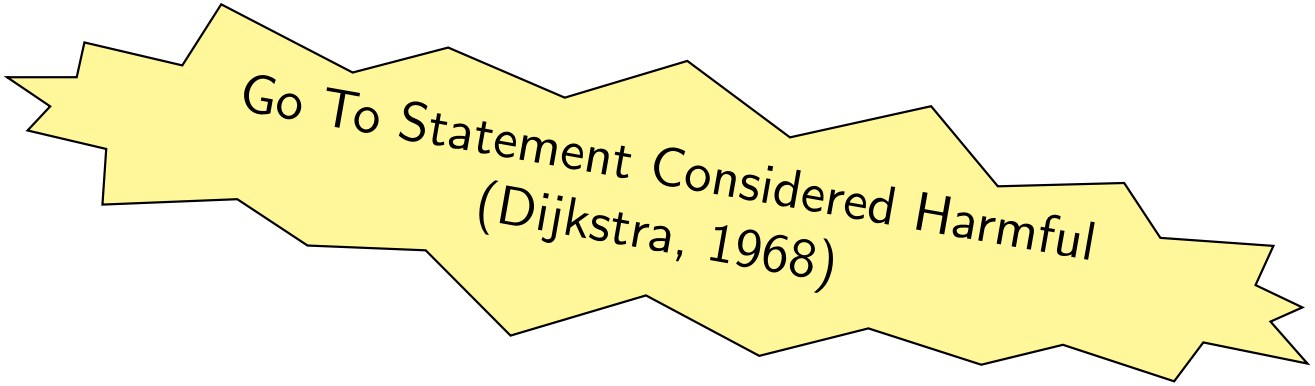
Why do we use functions?

- Organize code
- Re-use code

Functions

Why do we use functions?

- Organize code
- Re-use code



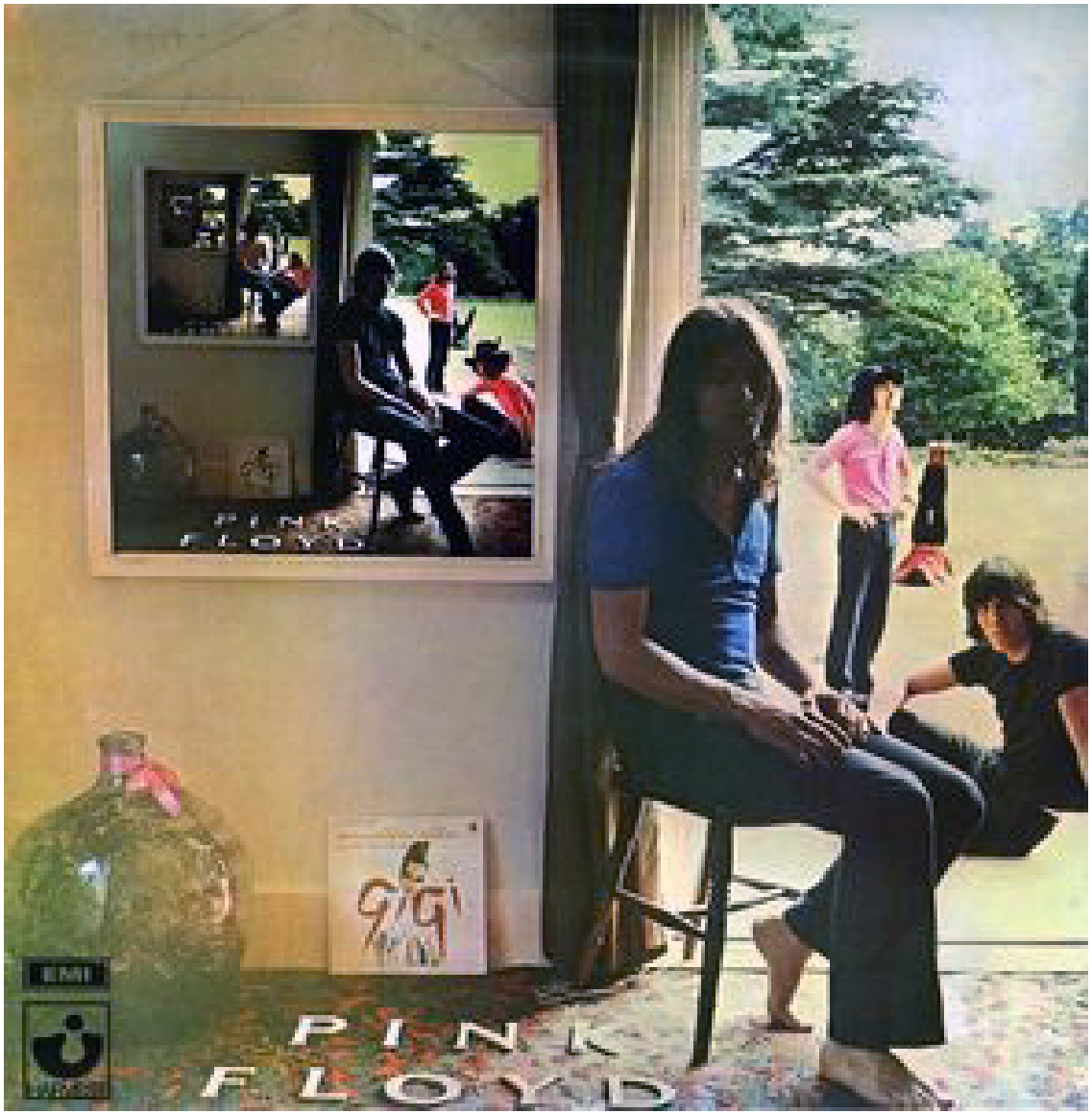
Functions

```
def callMe(input):
        b = 5;
        return input*b;

x = callMe(5);
print(x);
```

Functions

```
def callMe(input):
        b = 5;
        x = 10;
        return input*b;

b = 3;
x = callMe(5);
print(x);
print(b);
```

Recursive Functions



Recursive Functions

GNU

Recursive Functions

GNU

GNU = GNU's Not Unix!

Recursive Functions

```
def callMe(input):
        callMe(5);
        return 0;

x = callMe(5);
print(x);
```

What will be printed?

Recursive Functions

```
def callMe(input):
        print(input);
        if (input >= 0):
                callMe(input - 1);
        return 0;

x = callMe(5);
print(x);
```

Outline

Fibonacci numbers

Fibonacci numbers

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144...

Each number is the sum of the two previous ones.

RSA Encryption

- Public key: $n = p \times q$, where p and q are prime
- Private key: "Easy" calculations using p and q

If you can factorize numbers, you can break the security of the Internet!

RSA Encryption

- Public key: $n = p \times q$, where p and q are prime
- Private key: "Easy" calculations using p and q

No one knows how to do it quickly...

Assignment

1. Write a program that, given a number x, prints the xth Fibonacci number (2%)
2. Write a program that, given a number y, prints y's prime factors (2%)
3. According to Dijkstra, 1968, why Go To statements are not advisable in a high level programming language? (1%)

Assignment

- Deadline: **9am** the following Monday
- Submit your source code
- Submit the output of test cases
- Submit your reply to Question 3
- Write a short reflection about your learning/problem solving experience

Rules of the Game

- Discussions are allowed
- Searching for algorithms online is allowed (e.g., pseudo-code)
- **Copying and pasting full Python code directly IS NOT allowed**
  - Replacing variable names is still copying and pasting!
  - Just changing the text output is still copying and pasting!
- Everything will do must be reported in your short reflection

Thank you!

l.marcolino@lancaster.ac.uk
http://www.lancaster.ac.uk/staff/sorianom/