Introduction to Programming Discussion 1

Andrew Jianzhong Liu

ShanghaiTech University SIST

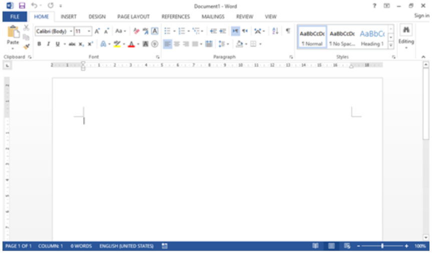
liujzh@shanghaitech.edu.cn

December 3, 2016

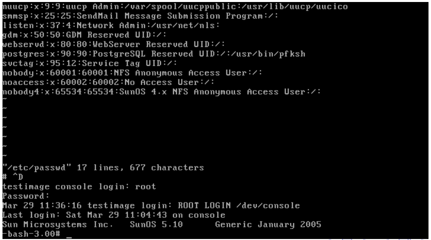
Overview

What is a Program?

- This one?



- Or this one?



```
nuucp:x:9:9:uucp Admin:/var/spool/uucppublic:/usr/lib/uucp/uucico
smmsp:x:25:25:SendMail Message Submission Program:/:
listen:x:37:4:Network Admin:/usr/net/nls:
gdm:x:50:50:GDM Reserved UID:/:
webservd:x:80:80:WebServer Reserved UID:/:
postgres:x:90:90:PostgreSQL Reserved UID:/:/usr/bin/pfksh
svctag:x:95:12:Service Tag UID:/:
nobody:x:60001:60001:NFS Anonymous Access User:/:
noaccess:x:60002:60002:No Access User:/:
nobody4:x:65534:65534:SunOS 4.x NFS Anonymous Access User:/:
~
~
~
~
~
~
~

"/etc/passwd" 17 lines, 677 characters
# ^D
testimage console login: root
Password:
Mar 29 11:36:16 testimage login: ROOT LOGIN /dev/console
Last login: Sat Mar 29 11:04:43 on console
Sun Microsystems Inc.   SunOS 5.10      Generic January 2005
-bash-3.00# _
```

The Definition of a Program

Computers

A computer is a tool for solving problems with data.

Program

A program is a sequence of instructions that tell a computer how to do a task.

A Sample Program ... to Make a Sandwich

Ingredients

- Two pieces of bread
- Ham
- Lettuce
- Ketchup

Procedure

1. Heat the bread in an oven for 2 minutes with moderate power.
2. If the bread is not warm enough, repeat step 1.
3. Put some ham on one piece of bread and some lettuce on the other.
4. Apply ketchup to the ham.
5. Join the two pieces by the side with the contents.
6. Bon appetite!

A Sample Program ... to print out the first N natural numbers

```
{The "Ingredients" of this program}
var i:integer;
var n:integer;

{The "Procedure" of this program}
begin
  read(n);
  for i:= 0 to n-1 do
    writeln(i);
end.
```

sample1.pas

The Same Program in Python.

Try it yourself!

```
#!/usr/bin/python3

# The same ingredients
i = 0
n = 0

# And the same procedures
n = int(input())
for i in range(n):
    print(i)
```

sample2.py

Concepts

- **Execution** When a computer follows the instructions in a program, we say it executes the program.
- **Variables** A variable is a storage location and an associated symbolic name which contains some known or unknown quantity or information, a value.
- **Control Structure** A control structure is a block of programming that analyzes variables and chooses a direction in which to go based on given parameters.
- **Data structures** A data structure is a particular way of storing and organizing data in a computer so that it can be used efficiently.
- **Syntax** The syntax of a programming language is the set of rules that define the combinations of symbols that are considered to be correctly structured programs in that language.

Concepts in Action: Execution

Demonstration

1. Compiled Binaries
2. Scripts
3. Execution w/o OS

Concepts in Action: Variables and Control Structures

```c
#include <stdio.h>
int main(){
/* Two variables i and n of integer type */
  int i,n;
  scanf("%d",&n);
/* Control flow
 * for i initially at 0
 * when i < n
 * execute the statements in the block
 * add 1 to i and loop
 */
  for(i = 0; i < n; i++){
    printf("%d\n",i);
  }
  return 0;
}
```

sample3.c

What a Computer Understands

Depends on the **ISA** (Instruction Set Architecture). Generally called **Machine Code**.

Machine Code Example (MIPS)

opcode	rs	rt	rd	shamt	funct
000000	00001	00010	00110	00000	100000

  Meaning Add the values in registers 1 and 2 and place the result in register 6.

opcode	target address
000010	00000 00000 00000 10000 000000

  Meaning Jump to the instruction at address 1024.

Downside is **very** obvious: Difficult for programmers to interpret!

Machine Code for Humans: Assembly

Description

An assembly language (asm), is a low-level programming language for a computer, or other programmable device, in which there is a very strong (generally one-to-one) correspondence between the language and the architecture's machine code instructions.
i.e. Machine Code that Humans can read and understand easily.

Assembly Language Example

```
; Boot sector code offset: 0x7c00

  mov ah, 0x0e
  mov bl, string
  add bx, 0x7c00
print_next:
  mov al,[bx]
  cmp al, 0x0
  je inf_loop
  int 0x10
  add bl, 0x1
  jmp print_next
inf_loop:
  jmp $
string:
  db "Hello World",0
  times 510-($-$$) db 0
  dw 0xaa55
```

string.asm

High Level Languages

Description

A high-level programming language is a programming language with strong abstraction from the details of the computer.

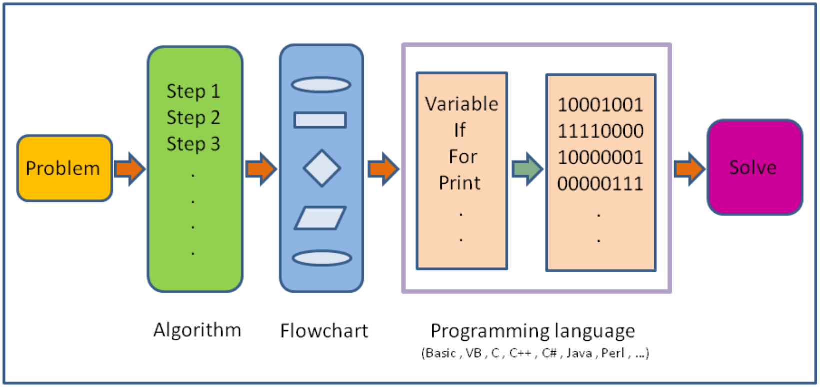
Examples

- **Compiled w/o Runtime**
  - C,C++
  - Pascal
  - *Rust*
- **Compiled w/ Runtime**
  - C#, Visual Basic .NET, Anything with .NET
  - Golang
- **Interpreted**
  - Python
  - Javascript

Comparison Between Programming Languages

Type	Corr. w/ M.C.	Difficulty	Compile Time	Runtime
Assembly	High	High	Low	Low
Compiled w/o R.	Medium	Medium	High	Low
Compiled w/ R.	Low	Low	Medium	Medium
Interpreted	Very Low	Very Low	Low to None	High

A Program from Conception to Implementation



Simple Procedural Programs

Procedures

Procedures contain a series of computational steps to be carried out.

Simple Procedural Programming

These programs are simply one large block of instructions. Loops may appear but no functions, classes are apparent.

Usage

Simple problems without requiring modular programming.

Simple Procedural Programs Cont.

Sample Problem 1: Fibonacci Numbers

Description

Fibonacci Numbers $\{f_n\}$: $f_0 = 0, f_1 = 1$. For any $n \geq 2$, $f_n = f_{n-1} + f_{n-2}$. For any n, print out the corresponding Fibonacci number f_n.

Input

One line n, where n is an integer between $[1, 45]$.

Output

One line f_n.

Sample Input and Output

Input: 5
Output: 5

Fibonacci Numbers Sample Code 1

```
a = 0
b = 1
tmp = None
c = int(input())
if c== 0:
  print(0)
else:
  for i in range(c-1):
    tmp = a+b
    a = b
    b = tmp
  print(b)
```

fibonacci.py

Exercise Problem 1: Series

Description

We know a series $S_n = 1 + \frac{1}{2} + \frac{1}{3} + ... + \frac{1}{n}$. Apparently for any number K, there exists an n that satisfies $S_n > K$. Find the smallest n for a given number K.

Input

One line, an integer K.

Output

One line, an integer n.

Sample Input and Output

Input:1
Output:2

Answer Code Please!

We would appreciate it if you would like to show your code!

Control Flow Outline

Description

Control flow (or alternatively, flow of control) is the order in which individual statements, instructions or function calls of an imperative program are executed or evaluated.

Primitives

- Labels and Goto
- Sequences
- Subroutine (VB.NET: Sub keyword)

Choice

- If ... else statements
- Switch ... case statements

Control Flow Outline Cont.

Loops

- Simple Infinite Loops (Rust loop block)
- While loops
- For loops (Syntax sugar)

Others

- Early exit: break
- Continuation: continue
- Exceptions
- Non-returning macros *(Rust)*

Sample Problem 2: Bubble Sort

Description

Bubble sort is an easy way of sorting an array. Simply traverse the array n times where n is the length of the array and compare adjacent elements. If they are of incorrect order, then swap the two elements.

Input

First line: An integer n, $1 \leq n \leq 20$. For the next n lines, there will be an integer each.

Output

One line, n integers in ascending order.

Sample Input and Output

Input: 3 5 2 7
Output: 2 5 7

Data Structures Needed

Arrays

An array is a data structure consisting of a collection of elements (values or variables), each identified by at least one array index or key.

Python implementation: List

```python
arr = [] # Initialization of an empty array
arr.append(1) # Append an integer of value 1 to the end
arr.pop(0) # Remove and return the element from the array
    with index 0
```

arr.py

Bubble Sort Sample Code

```
n = int(input())
nums = []

for i in range(n):
  nums.append(int(input()))

for i in range(n):
  for j in range(1,n - i):
    if(nums[j-1] > nums[j]):
      tmp = nums[j-1]
      nums[j-1] = nums[j]
      nums[j] = tmp

for i in range(n):
  print(nums[i],end = ' ')
print()
```

bubblesort.py

Code Reuse

Scenario

What if I needed to sort in many parts of my program or project?

Example

In an operating system, we need to sort files in the file explorer according to either name, date, size and/or other properties. In addition, in the utility Task Manager, we would also need to list the processes by process ID or name.

Subprograms

Theory

In general, complex things break more often than simple things. It is common to solve big, complicated problems by breaking them down into a group of smaller, simpler problems.

Subprograms

We can write several small pieces of a program that each solve part of the problem and, then, combine them together, in some specific way, to solve the entire problem.

Note: Subprograms go by different names depending on the programming language being used. We will mainly discuss functions.

Bubble Sort Algorithm as a Function

```python
def bubblesort(nums,n):
  for i in range(n):
    for j in range(1,n-1):
      if(nums[j-1] > nums[j]):
        tmp = nums[j-1]
        nums[j-1] = nums[j]
        nums[j] = tmp
  return nums

n = int(input())
nums = []
for i in range(n):
  nums.append(int(input()))

nums = bubblesort(nums,n)
for i in range(n):
  print(nums[i],end = ' ')
print()
```

bubblesortfunc.py

Function Fundamentals

Declaration

- Return type
- Function Name
- Parameters

Calls

Use the function name and pass in parameters. Can be fun to use, can be used to build a program like LEGOs.

Parameters / Arguments

Parameters allow you to pass values to the procedures and functions that you declare, you can pass all sorts of data as parameters and you can pass as many as you like. Note: Parameters are not exactly equal to Arguments.

Function Exercises

Try writing these functions below:

1. A function which takes in an array of numbers and finds the smallest one (min function)
2. A function which swaps two numbers in place (No intermediate value needed!)

Exercise 1: GPA Calculation

Description

Yuan Shen is about to graduate from college. However, due to a power outage, the grades and GPA (Grade Point Average) of his college career has disappeared from all the databases. However he has a list of the scores and academic credits of the courses he has taken. To find a job, he will need to provide his employers with his GPA. Can you help him calculate his GPA so that he can graduate and get employed?

$$GPA = \frac{\sum(academic\_credits * grades)}{\sum(academic\_credits)} \tag{1}$$

Exercise 1: GPA Calculation Cont.

Input

First line: An integer n, $1 \leq n \leq 20$. For the next n lines, there will be an integer for grades and another integer for academic credits.

Output

One line, a floating point number for his GPA.

Sample Input and Output

Input:

```
2
4 4
3 2
```

Output: 3.66666666

Exercise 2: Plane Ticket Discounts

Description

The actual amount of money one has to pay for a plane ticket is the listed price times the discounts at the time of purchase, or $P_a = P_0 * d$, where P_a is the actual price, P_0 is the listed price and d is the discount ratio.

Input

One line, two numbers. The first an integer P_0, the second a floating point number d.

Output

One line, one floating point number P_a.

Exercise 3: Area Calculation

Description

The area of a circle is calculated by $A = \pi * r^2$. If we have two circles with the larger one encompassing the smaller one, calculate the area of the diagram belonging to the larger circle but not in the smaller circle.

Input

One line, two integers r_1, r_2 and $0 < r_2 < r_1 < 100$.

Output

One line, one floating point number A.

Sample Input and Output

Input: 2 1
Output: 9.4247

Exercise 4: Greatest Common Divisor & Least Common Multiple

Description

Greatest Common Divisor or GCD of a and b is such an integer value c that both of a, b are divisible by it (e.g. leave no remainder) - and also c is the largest possible. For example gcd(20, 35) = 5 and gcd(13, 28) = 1.
Euclid's algorithm is quite simple - we keep on subtracting smaller value (of a and b) from larger - and repeat this operation until values become equal - this last value will be gcd.
For example: $GCD(20, 35) = 5$. (Demonstration)
Least Common Multiple (or LCM) of a and b is such an integer d that it is divisible by both of them (and is the smallest of all possible). It can be found with the following rule: $lcm(a, b) = a * b/gcd(a, b)$

Exercise 4: Greatest Common Divisor & Least Common Multiple Cont.

Input

Two integers a, b.

Output

Two integers, c the GCD and d the LCM.

Sample Input and Output

Input: 2 3
1 6

Text Editors

- Vi, Vim, GVim, NeoVim
- Emacs
- Atom, Sublime Text
- Visual Studio Code, Notepad++

Operating Systems

- Windows
- OS X
- Debian/Ubuntu/Mint/Deepin
- RHEL/Fedora
- CentOS
- Arch, Antergos
- FreeBSD, NetBSD, OpenBSD

Information on Programming

- Reddit
- StackOverflow
- MSDN
- Wikipedia
- TIOBE

Free and Open-source Software

Definition

Free software is computer software distributed under terms that allow users to run the software for any purpose as well as to study, change, and distribute the software and any adapted versions.

Licenses

- GPL
- BSD
- MIT
- Creative Commons
- Apache

Any Questions?

Muchas Gracias y Buenas Noches