

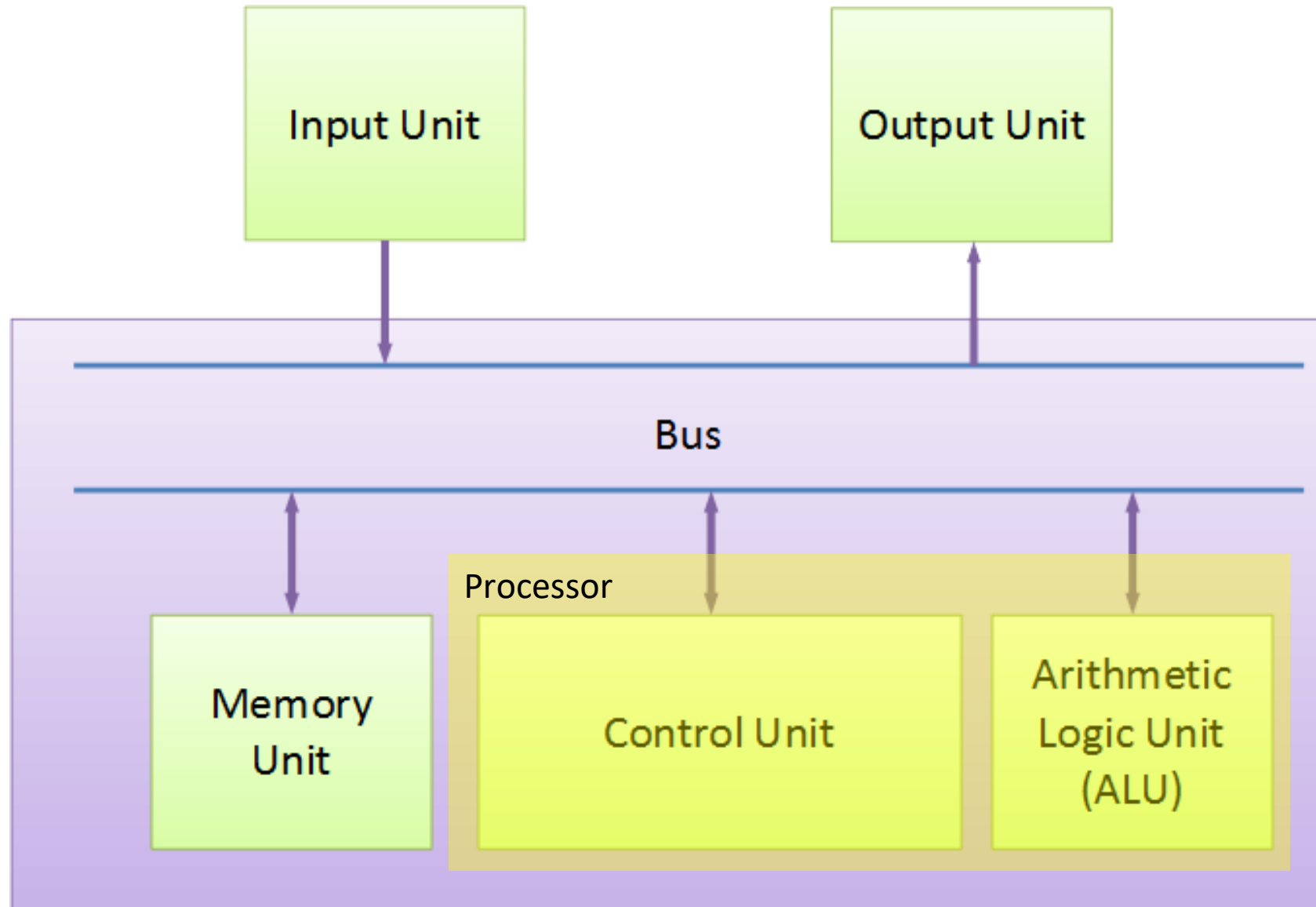
CSAI 101: Fundamentals of Programming and Computer Science

Lecture 01: First Program

Agenda

- Computer hardware and software
- How to write a program?
- Building our first program
- Memory
- Getting input / Displaying output
- Arithmetic expressions
- Sample programs

Computer Hardware



Computer Hardware

- **Input Unit:** accepts inputs from users, translates it, and transmits it to the processor (e.g. keyboard, joystick, microphone and mouse)
- **Output Unit:** display processed information to the outside world (e.g. monitors, printers (2D and 3D), speakers, and projectors)
- **Arithmetic and Logic Unit:** calculates arithmetic and logical expressions. It reads the operands from memory, perform the operation and store the resultant value in memory.
- **Memory Unit:** stores programs and data (two classes primary and secondary)
- **Control unit :** coordinates the operation of all other units. It is the nerve center that sends control signals to all units and senses their status .

Software Program

- A computer can perform a specific set of instructions, one at a time.
 - Get **input** (from keyboard to memory)
 - Display **output** (from memory to screen)
 - Compute an arithmetic expression (*, /, +, -)
 - Compute a logical expression and branch selection
 - Repeat instructions
- A software program is a sequence of such instructions

How to write a software program?

1. Read the **problem** statement, and identify
 - The **input** and its range
 - The **output**
 - The **relationship** between the input and the output (how to compute the output) [Comprehend]
2. Write your thoughts as a sequence of steps. [Algorithm]
3. Convert these steps to Code. [Program]
4. Test your code and compare your program result against a human result. [Testing]

Writing a Computer Program: The Role of Algorithms

- An algorithm is a set of steps that describes how a task can be performed (informal definition).
- Fundamental concept of computer science
- The word is derived from the famous mathematician Muḥammad ibn Musa al-Khwarizmi, one of the first Directors of the House of Wisdom (a major public academy and intellectual center), in Bagdad (9th century).
- Began as a subject in mathematics. The goal was to find a single set of directions that describes how all problems of a particular type could be solved.
- **“If no algorithm exists for solving a problem, then the solution of that problem lies beyond the capabilities of machines.”**



1. From the book “*Computer Science: an overview*” by J. Glenn Brookshear

Algorithms: Formal Definition

- An algorithm is an **ordered** set of **unambiguous**, **executable** steps that defines a **terminating** process.
 - **Ordered:** Steps are executed in a sequence. (In parallel algorithms, we may have more than one execution path)
 - **Unambiguous:** A single and unique interpretation. The information must be sufficient to determine uniquely and completely the required actions.
 - Branch if the value of x is large enough. Is it unambiguous?
 - Some algorithms (nondeterministic algorithms) do not conform to this restriction.
 - **Executable:** A step that can be done.
 - Make a list of all the positive integers. Is it executable
 - **Terminating:** the execution must lead to an end.
 - There are, however, some meaningful non-terminating processes such as monitoring the vital signs of a patient.

Algorithm Representation

- **Pseudocode:**

- An intuitive notational system in which ideas can be expressed informally.
- Less formal than target programming language.
- Flexible, less complex, and easy to understand compared to formal programming languages.

- **Flowchart:**

- A graphical representation of an algorithm.
- Provides a simple way to visualize transitions and analyze the execution paths.

Pseudocode: Examples

- Write an algorithm to calculate the area of a circle
- Determine inputs, outputs, and relation
- Step 1: Get Radius from user
- Step 2: Set $\text{Pi} = 3.14$
- Step 3: Calculate $\text{Area} = \text{Pi} * \text{Radius} * \text{Radius}$
- Step 4: Print Area

Pseudocode: Examples

- Write an algorithm to convert temperature from Fahrenheit to Celsius
- Determine inputs, outputs, and relation

$$T(^{\circ}\text{C}) = (T(^{\circ}\text{F}) - 32) \times 5/9$$

- **Step 1:** Get Temp in Fahrenheit (F) from user
- **Step 2:** Calculate $C = 5/9 * (F-32)$
- **Step 3:** Print C

Exercise

- Find X^2 and X^3 of any given number X
- Determine inputs, outputs, and relation
- **Step 1:** Get X from user
- **Step 2:** Set $XSquare = X * X$
- **Step 3:** Set $XCube = XSquare * X$
- **Step 4:** Print $XSquare$. $XCube$

Memory

- Memory is divided into bytes
- Each byte has its own address represented by binary numbers
- 4 or 8 bytes form a memory location to store integer or floating (Ex 10.2345) numbers
- Software programs use letters and numbers to name the memory location.

x	15
y	12
sum	45
A	2
A1	-3
numOfBus	4

Memory Location Name (Variables)

- **Variable:** a named space for storing a value radius
- Valid names:
 - Must start with a letter or '_',
 - Can contain any combination of letters, digits, or '_'.
- Use meaningful variable names (avoid reserved words).
- Python (as well as most others) is case sensitive (area and Area are different)
- **Exercise**

Xyz	3xyz	x-y	ab2cd	a12345	myvar	AbCdE1234
-----	------	-----	-------	--------	-------	-----------

Variable Types

- Integer:

- `x = 4`

- `y = x * 2`

- Using substitution, `y = ?`

- `type(x)`

- `type(y)`

- `type(x * 2)`

- Again follow substitution

Variable Types

- Float/decimal:

- `x = 4.0`

- `y = x * 2`

- Using substitution, `y = ?`

- `type(x)`

- `type(y)`

- `type(x * 2)`

Variable Types

- String:

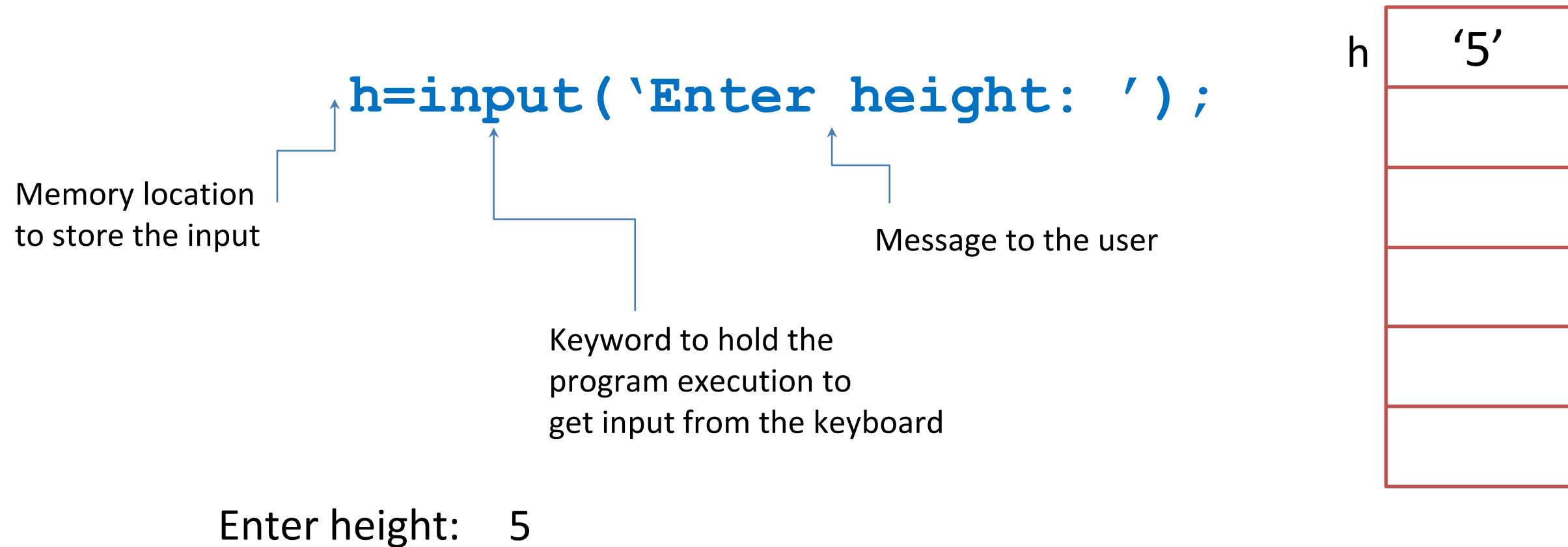
- `x = 'A'`
- `x = 'abc'`
- `x = '4'`
- `x = '4.0'`
- `x*2`
- `x + '3'`
- `type(x + '3')`
- `x + 3`

Variable Types

- Conversions:

- `int('4')`
- `int(4.5)`
- `float('4.2')`
- `float('4')`
- `str(4.2)`
- `'Area=' + str(4.2)`

Get Input from User



Get Input from User

```
h=int(input('Enter height: '))
```

int('5')=5

Returns (substitutes in its place) the string '5'

h

5

Enter height: 5

?

Write a Python statement to take the circle radius from the user and store it in 'R'.

Display to Screen

`print(h)`

Keyword to display
to the screen

Memory location
to display

? 5

`print(a)`

? 102

h	5
a	102

Display to Screen

```
print('Height = ' + str(h))
```

❓ Height = 5

```
print('Height =', h)
```

❓ Height = 5

Note regarding separating by a comma:

- No type conversion needed
- Automatically inserts space between the elements

```
print('Height =', h, 'and area =', a)
```

❓ Height = 5 and area = 102

h	5
a	102

Arithmetic Expressions

Operations

****** Exponentiation (5**2 is 25)
*****, **/**, **//** Multiplication, division, and floor division
+, **-** Addition and subtraction

Examples

```
x=4; y=3.5; z=2  
a = x + y - 5 / (3+z)  
b = a + x / 2
```

(semicolon can be used to separate multiple lines)

x	4
y	3.5
z	2
a	6.5
b	8.5

Operations are executed only one at a time according to their precedence. We will study it further in future lectures

How to write a software program?

1. Read the **problem** statement, and identify
 - The input and its range
 - The output
 - The relationship between the input and the output (how to compute the output) [Comprehend]
2. Write your thoughts as a sequence of steps. **[Algorithm]**
3. Convert these steps to Code. **[Program]**
4. Test your code and compare your program result against a human result. **[Testing]**

Calculate Area of a Rectangle

(Problem ? Algorithm)

- How a computer can solve this problem?
- I get the values of the height and the width from you and I store them in my memory.
 - ? Get height as h
 - ? Get width as w
- Then I multiply them in my brain and store the result in my memory
 - ? Calculate area = $h * w$
- And inform you about the resultant value.
 - ? Print area

Calculate Area of a Rectangle

(Algorithm ? Program)

Get height as h

```
h = int(input('Enter height: '))
```

Get width as w

```
w = int(input('Enter width: '))
```

Calculate area = $h \times w$

```
area = h * w
```

Print area

```
print(area)
```

Calculate Area of a Rectangle

(Program and Testing)

```
h=int(input('Enter height: '))  
w=int(input('Enter width: '))  
area = h * w  
print(area)
```

Enter height: 5
Enter width: 4
20

h	5
w	4
area	20

Calculate Area of a Circle

(Algorithm, Program, Testing)

1. Get radius r

2. Calculate $\text{area} = \pi * r^2$

3. Print area

```
import math
```

```
r = int(input('Enter radius:'))
```

```
area = math.pi*r**2
```

```
print(area)
```

Enter radius: 5
78.5398

radius	5
area	78.5398

Calculate Number of Buses to Transfer Students

(Algorithm)

1. Get number of students as numS
2. Get the bus capacity as busC
3. Calculate num of buses as
$$\text{numB} = \text{numS} / \text{busC} \text{ (round up)}$$
4. Print numB

Calculate Number of Buses to Transfer Students

(Program)

```
import math
numS = int(input('Enter number of students:'))
busC = int(input('Enter bus capacity:'))
numB = numS/busC          ? numB = math.ceil(numS/busC)
print(numB)
```

If numS=60 and busC = 50 then numB = 60/50 = 1.2 ???

But it should be 2

$\text{math.ceil}(1.2) = 2$

$\text{math.floor}(1.2) = 1$

ceil rounds up to nearest integer

floor rounds down to nearest integer

Calculate Number of Dozens and Remainder

(Algorithm)

1. Get a number as N
2. Calculate dozen = $N/12$ (no fraction)
3. Calculate the remainder (r) of dividing N by 12
4. Print dozen
5. Print r

Calculate Number of Dozens and Remainder

(Program)

```
import math
N = int(input('Enter a number'))
dozen = N/12
r = ??
print(dozen)
print(r)
```

☐ dozen = N // 12

☐ r = N % 12

N // 12 is the same as math.floor(N / 12)