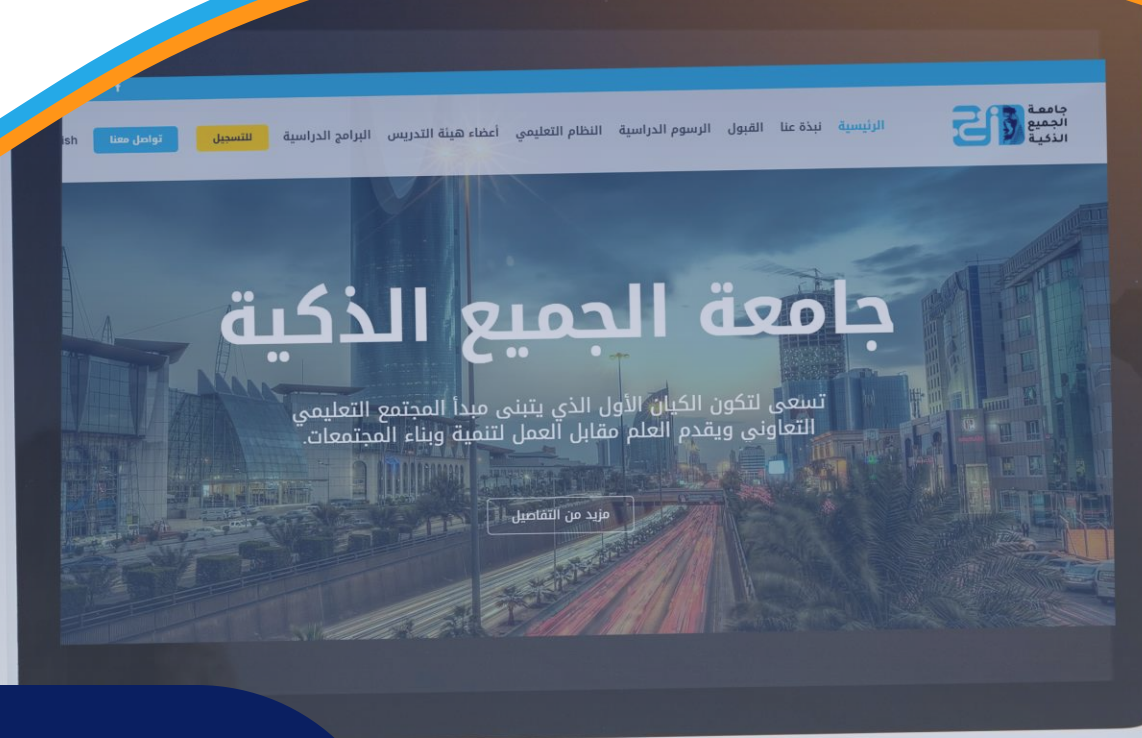




جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY



Data Types and Input-Output Operations
Prepared by Dr. Hamada I. AbdulWakel
Lecture 2

CYB550
Programming Foundations In
Python

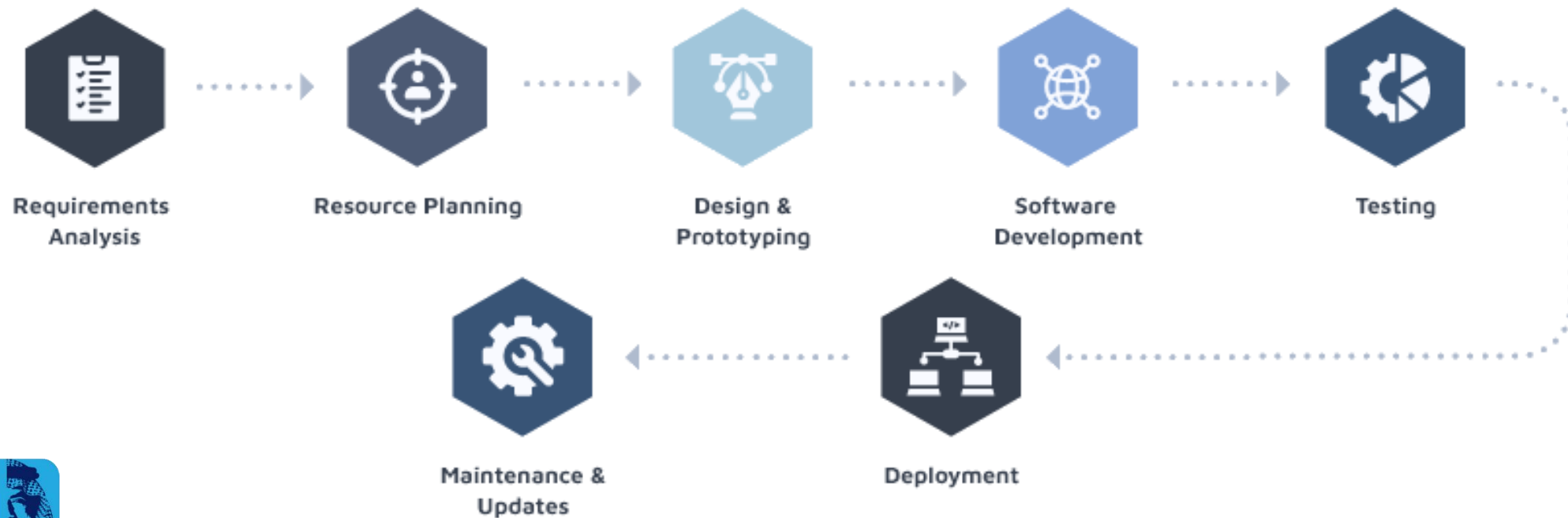
- Problem solving is a process of information processing.
- Converting information processing into algorithm.
- Python programming language.
- Requirements for python.

- Describing the basic phases of software development.
- Dealing with input and output operations in python.
- Knowing the data types and using variables.
- Constructing arithmetic expressions.

Lecture 2

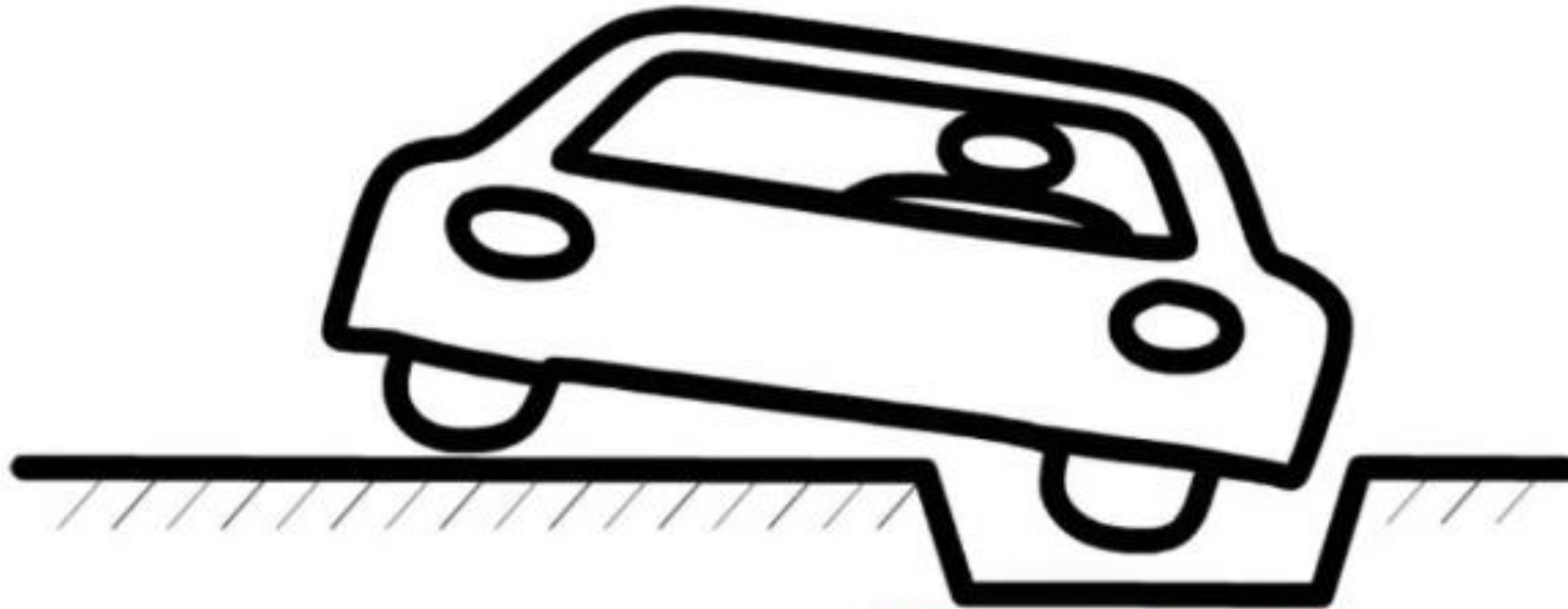
THE SOFTWARE DEVELOPMENT PROCESS

- **Software development:** process of planning and organizing a program.
- Software Development Life Cycle (**SDLC**).



Lecture 2

SDLC APPROACHES: THINK & SOLVE



Pothole

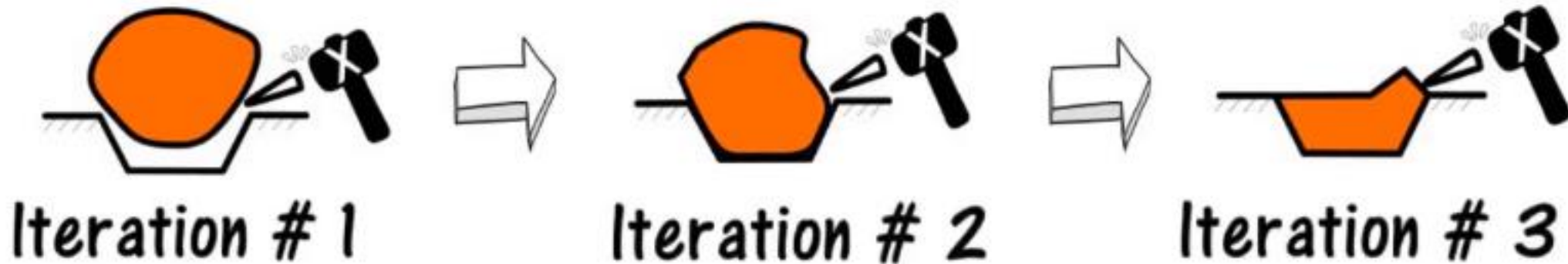


جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

WATERFALL APPROACH



ITERATIVE APPROACH



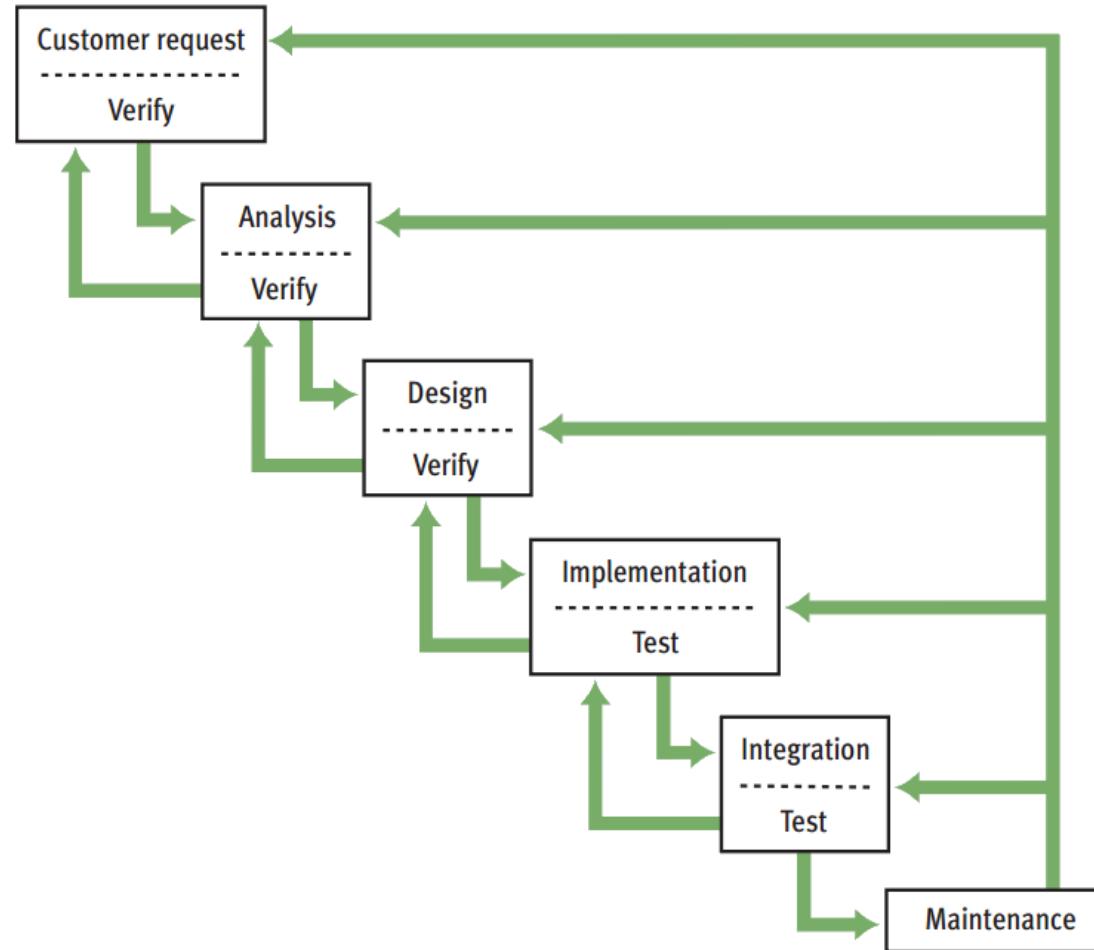
INCREMENTAL APPROACH



- There are different approaches that could be followed:
 - The Waterfall Model.
 - The iterative Model.
 - The incremental Model.

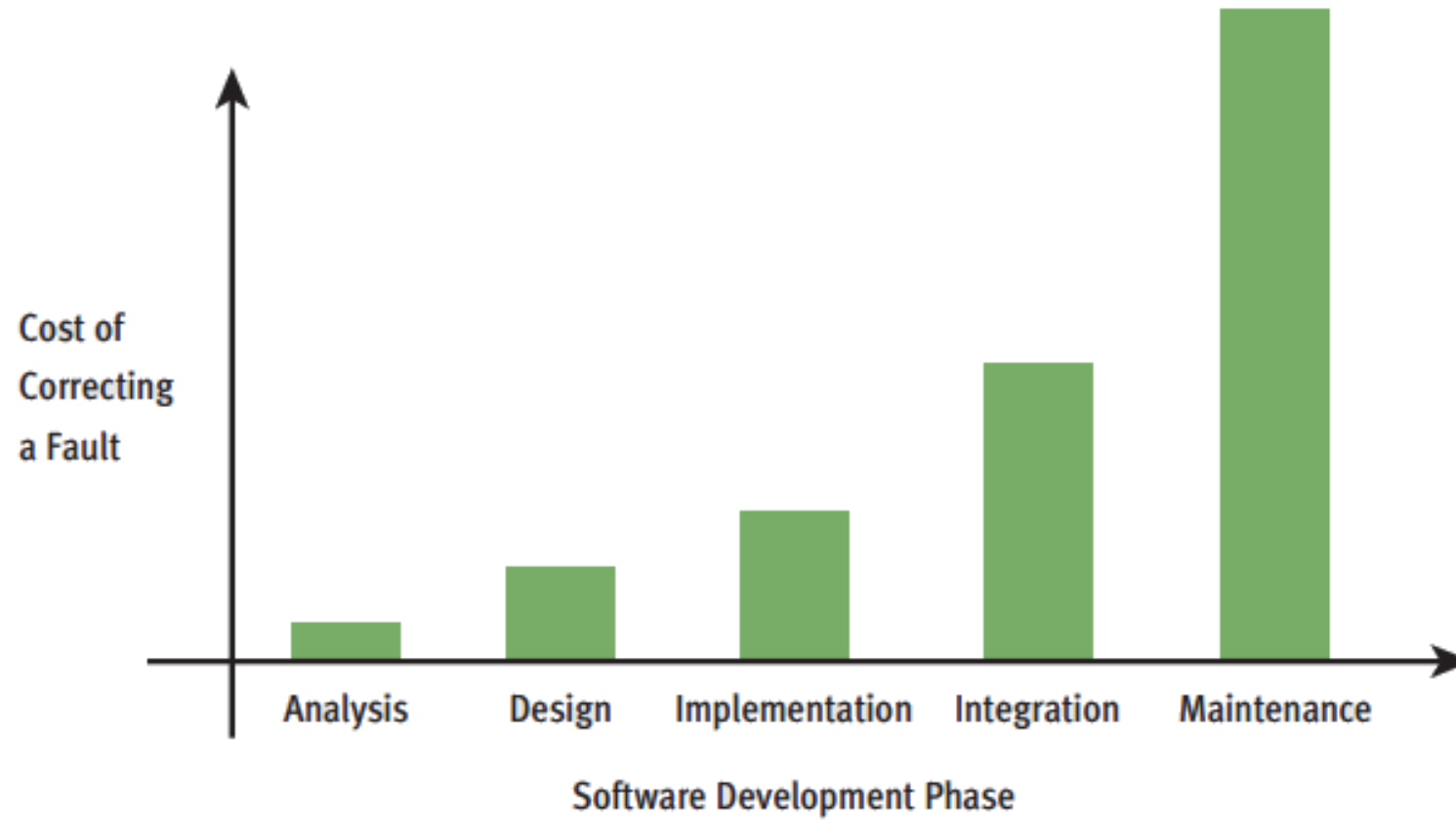
Lecture 1

SDLC APPROACHES: WATERFALL MODEL



Lecture 1

COST OF CORRECTING A FAULT



- Each year nearly everyone with an income faces the unpleasant task of computing his or her **income tax return**.

- The customer requests a program that computes a person's income tax.
 - Gross income required.
 - Flat tax rate 20%.
 - \$10000 standard deduction.
 - For each dependent, there is a \$2000 deduction.

Enter the gross income: *150000.00*

Enter the number of dependent: *3*

The income tax is **\$26800.00**

- An English-style algorithm is called **Pseudocode**.
 - Input the gross income and number of dependents.
 - Computer the taxable income using the formula
$$\text{Taxable income} = \text{gross income} - 10000 - (2000 * \text{number of dependent})$$
 - Compute the income tax using the formula
$$\text{Tax} = \text{taxable income} * 0.20$$
 - Print the tax

Lecture 2

CASE STUDY: INCOME TAX CALCULATOR IMPLEMENTATION

Cont'd

Python code later

- **Error types:**
 - **Syntax error:** occurred due to the mistakes while typing or do not follow the syntax of the specified programming language.
 - **Run-time error:** When the program is running, and it is not able to perform the operation.
 - **Logical error:** an error that leads to an undesired output.
- May use a **test suite** to test if program is correct.

CASE STUDY: INCOME TAX CALCULATOR
TESTING

Cont'd

NUMBER OF DEPENDENTS	GROSS INCOME	EXPECTED TAX
0	10000	0
1	10000	-600
2	10000	-1200
0	20000	2000
1	20000	1400
2	20000	800

- Almost all programs have solutions that needed to be displayed to the user through outputting data from the program.
 - Use the print() function.
 - help(print)
 - print("Hello", "Ahmed")
 - print("Hello", "Ahmed", sep="000")
 - Print("Hello", "Ahmed", end="1111")

```
print("In 2007.")  
print("Micaela paid $120,000 for her house.")  
print("Micaela' house at 24 East Maple Street.")
```

```
print("In 2007.")  
print("Aly paid $120,000 for her house.")  
print("Aly' house at 24 East Maple Street.")
```

```
Person_name = "Ahmed"  
print("In 2007.")  
print(Person_name + " paid $120,000 for her house.")  
print(Person_name + " house at 24 East Maple Street.")
```

```
Person_name = "Ahmed"  
Money = "120000"  
print("In 2007.")  
print(Person_name + " paid $" + Money + " for her house.")  
print(Person_name + " house at 24 East Maple Street.")
```

```
Person_name = "Ahmed"  
Money = "120000"  
print("In 2007.")  
print(Person_name + " paid $" + Money + " for her house.")  
Person_name = "Aly"  
print(Person_name + " house at 24 East Maple Street.")
```

- Python is a type-less language.
 - `X = 10`
 - `X = "Hello world"`
- The variable name refers to the memory location where its value store.
 - `I = 10    J = 10`
 - `print (id(I), id(J))`



- A variable name must begin with either a **letter** or an **underscore** (_)
- It contains any number of letters, digits, or other underscores.
- Variable names are **case sensitive**;
 - WEIGHT = 50
 - Weight = 50.
- Variable name could be unicode, means you can have س as a variable name.

- Main data types:
 - **Strings**. A set of characters enclosed by " "
 - **Numbers**. A number written without ""
 - **Booleans**. True or False value.
 - **List**.
 - **Tuple**.
 - **Dictionary**.
- Use `type(var_name)` to know variable type.

- A string literal is a sequence of characters enclosed in single or double quotation marks.
 - Examples:
 - `Name = "Hamada"`
 - `Name = str("Ibrahim")`
 - `Desc = ' ' ' Hello I'm from "Egypt" ' ' '`
 - On python shell: `""" test test """`
 - Strings are Immutable
 - `text[2] = 'x'` not supported

- Escape sequences are **characters** used to express **special characters** as literals.

ESCAPE SEQUENCE	MEANING
<code>\b</code>	Backspace
<code>\n</code>	Newline
<code>\t</code>	Horizontal tab
<code>\\</code>	The <code>\</code> character
<code>\'</code>	Single quotation mark
<code>\"</code>	Double quotation mark

- A string functions:
 1. Use + and * with strings
 2. `text.lower()` vs. `text.upper()`
 3. `text.isupper()` vs. `text.islower()`
 4. `len(text)`
 5. `text[0]` vs. `text[0:3]`

- A string functions:
 1. `text.index("a")` vs. `index("asd")`
 2. `text.replace("old", "new")`
 3. `print(text.lower().islower())`
 4. `print("subtext" in "text")`

- A string functions:
 1. `var.isalnum()` vs. `var.isalpha()` vs. `var.isdigit()`
 2. `chr(65)` vs. `ord('A')`

- Python's **int** data type consists of the integers from (-2^{31}) to $(2^{31} - 1)$.
- When the value of an integer exceeds these limits, Python automatically uses the **long (L)** data type to represent it.
- Python uses **floating-point** numbers to represent real numbers.
- Complex numbers (e.g., $3 + 2j$) are also supported.

Name	Meaning	Example	Result
+	Addition	15+20	35
-	Subtraction	24.5-3.5	21.0
*	Multiplication	15*4	60
/	Float Division	4/5	0.8
//	Integer Division	4//5	0
**	Exponentiation	4**2	16
%	Remainder	27%4	3

Operator	Name	Example	Equivalent
<code>+=</code>	Addition assignment	<code>x += 4</code>	<code>x = x + 4</code>
<code>-=</code>	Subtraction assignment	<code>x -= 2</code>	<code>x = x - 2</code>
<code>*=</code>	Multiplication assignment	<code>x *= 5</code>	<code>x = x * 5</code>
<code>/*=</code>	Division assignment	<code>x /= 5</code>	<code>x = x / 5</code>
<code>//*=</code>	Integer division assignment	<code>x //= 5</code>	<code>x = x // 5</code>
<code>%*=</code>	Remainder assignment	<code>x %= 5</code>	<code>x = x % 5</code>
<code>**=</code>	Exponent assignment	<code>x **= 5</code>	<code>x = x ** 5</code>

- Numeric functions:
 - `bin(50)`, `oct(50)`, `hex(50)`
 - `int(0x32)`
 - `abs(-5)`
 - `pow(2, 3)`
 - `max(4, 5)`
 - `min(4, 5)`
 - `round(3.7)`

- A comment is a piece of program text that the interpreter ignores but that provides useful documentation to programmers.
 - Using # symbol.
 - `""" ..... """`

- Variables receive their initial values and can be reset to new values with an **assignment statement**.
 - Variable defining/initializing
 - Variable references.

`<var_name> = <expr>`

- Expressions provide an easy way to perform operations on data values to produce other data values.
- An arithmetic expression consists of operands and operators.
 - Operators:
 - -, *, /, +

- Rule of **priority**, that is some operator has highest priority than others.
 1. `()`
 2. `**`
 3. Unary operator
 4. `*`, `/`, `%`
 5. `+`, `-`
- Rule of **associativity**, evaluate expression of similar priority from left to right.
 - Ex: `2+5-1`

- Almost all programs interact with the user through giving data to the program to process it and display other data.
 - Use the input() function.
 - input()
 - input("Enter your name: ")
 - name = input("Enter your name: ")
 - Age = int(input("Enter your Age: "))

Lecture 2

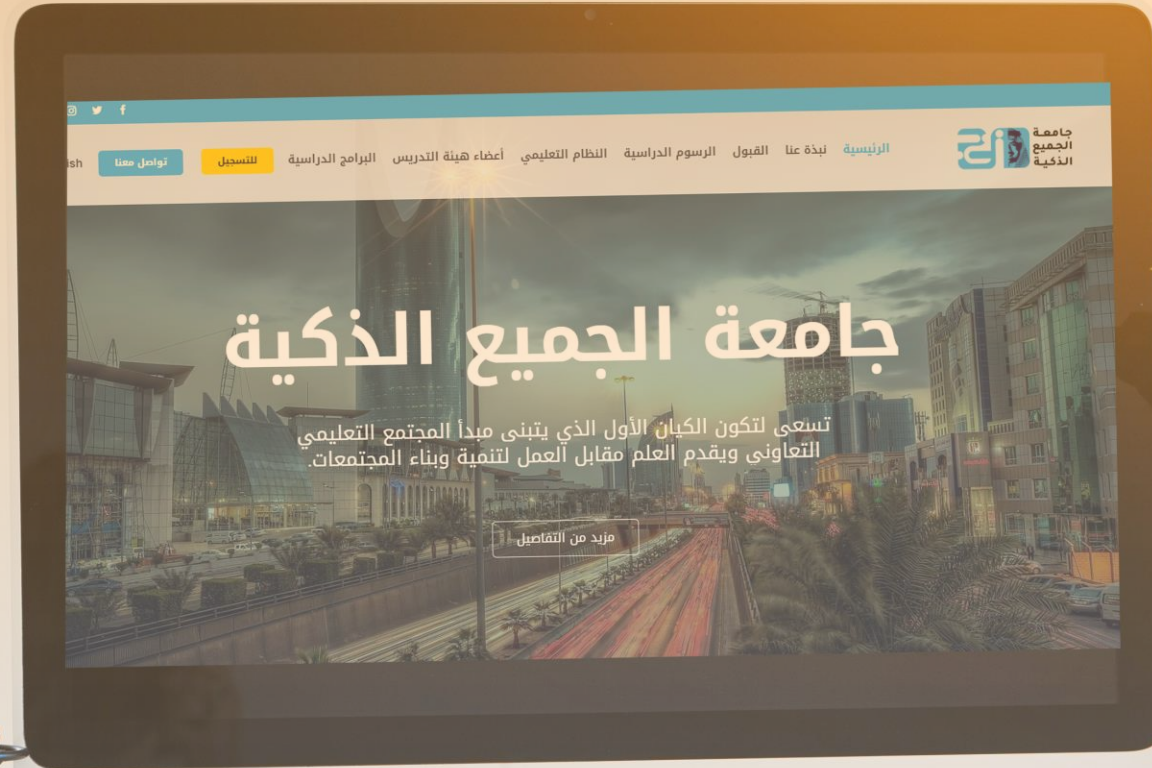
SIMPLE CALCULATOR IN PYTHON

```
Num1 = input("Enter the first number: ")  
Num2 = input("Enter the second number: ")  
  
Result = Num1 + Num2  
  
print(Result)
```

Lecture 2

CASE STUDY: INCOME TAX CALCULATOR IMPLEMENTATION

Try to implement it!!!



جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

THANK YOU