****

**Session 2**

**30/8/2022**

**Task 1:**

* **What are the components of OS?**
  1. Process management
  2. File management
  3. Network management
  4. Main memory management
  5. Secondary storage management
  6. I/O Device management
  7. Security management
  8. Command interpreter system
* **How Kernel works and manage processes?**

Manage RAM memory, so that all programs and running processes can work. Manage the processor time, which is used by running processes.

**Task 2:**

* How To Hide Password in Python?

Pip install maskpass

Import maskpass

Pwd=maskpass.advpass()

**Task 3:**

* How to write C++ in jupyter?

By using mybinder.org

**Task 4:**

* How to make do while in Python?

while True:

if condition:

break

for Example:

while True:

number = int(input(“enter a positive number: “))

print (number)

if not number > 0:

break

**Task 5:**

* How to write for loop and iterate times 2?

for x in (2\*\*p for p in range(1, 11)):

print(x)

* How to write for loop and iterate power 2?

for I in range(1,10):

I=I\*\*2

print(I)

**Task 6:**

* How to make infinite for loop in python?

Can do this by itertools package and import cycle

from itertools import cycle

a= [100]

for I in cycle(a):

print(I)

**Task 7:**

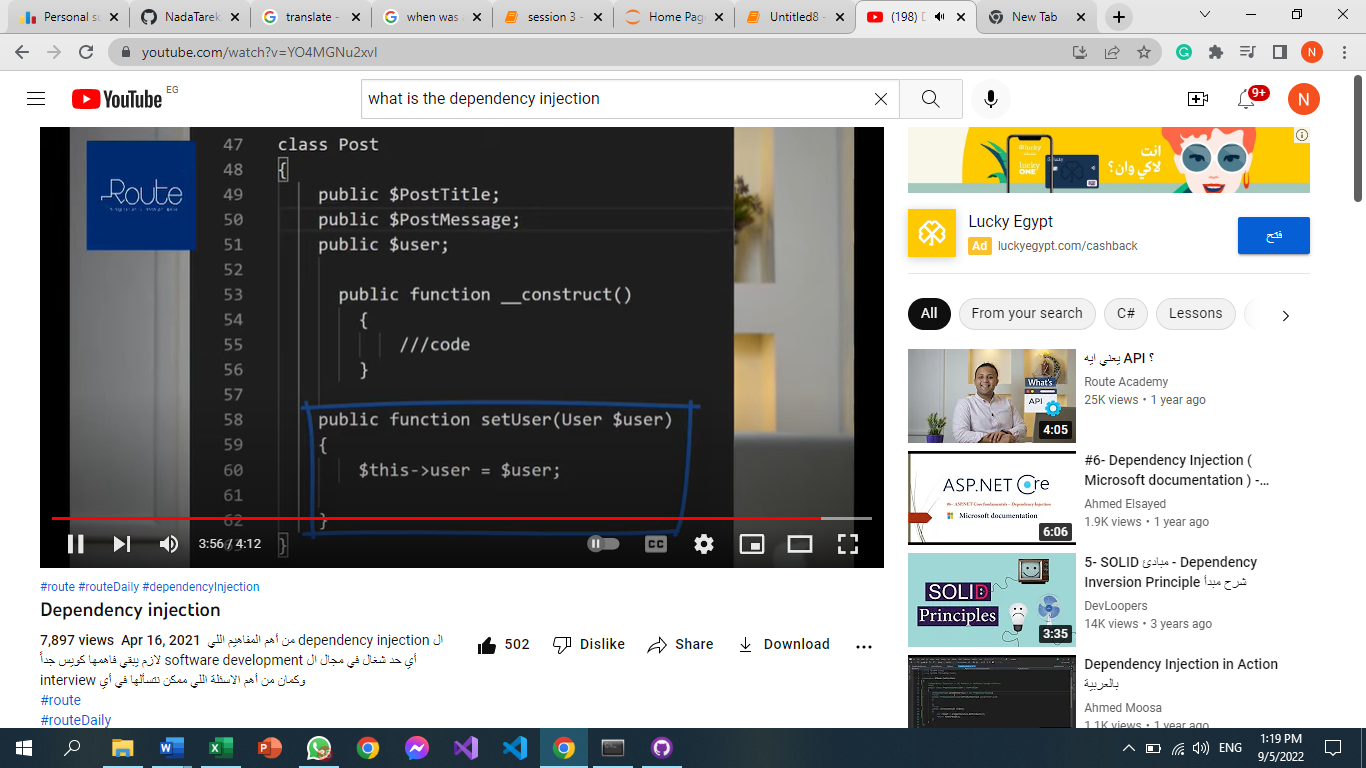
* What is the dependency injection?[[1]](#ref1)

Dependency : when a class depends on another class



Class post depends on class user

دلوقتى لما كلاس يكون معتمد على كلاس تانى لو غيرت حاجه فى الاساسى لازم اغير فى كل اللى معتمدين عليه ف احنا لو بنتعامل مع سيستم كبير دى كدا هتكون مشكله ف من هنا ظهر المفهوم دا يعنى بحاول اظبط انى لو عملت اى تغير يبقى ميأثرش على الباقى:



In object-oriented programming software design, dependency injection is the process of supplying a resource that a given piece of code requires. The required resource, which is often a component of the application itself, is called a dependency.[[2]](#ref2)

* **Advantages:**

 basic benefit of dependency injection is decreased coupling between classes and their dependencies.

**Task 8:**

* What are the Rules of clean code?

-Clean code rules by Robert C. Martin:

* + Names Rules:
    1. Choose descriptive names.
    2. Makes meaningful.

**Example**:

Bad: var d; // elapsed time in days

Good: var elapsedTimeInDays.

var daysSinceCreation.

var daysSinceModification.

* + 1. Use pronounceable names.
    2. Use searchable names.

**Example:**

Bad: if (student.classes.length < 7) {

// Do something}

**Good:** if (student.classes.length < MAX\_CLASSES\_PER\_STUDENT) {

// Do something}

* + Functions Rule:
    1. Small
    2. Do one thing
    3. Use descriptive names
    4. Prefer fewer arguments
  + Comments Rules
    1. Always try to explain yourself in code
    2. Don’t be redundant
    3. Don’t comment out code
    4. Don’t add obvious noise

**Task 9:**

* **Why recursion faster than iterative?**

**Task 10:**

* **What are the threads?**[**[3]**](#ref3)
  + A Thread is the basic unit of CPU utilization
  + A process starts with a single thread and later can create more threads from any of its threads
  + All threads of the same process have access to its memory
  + A thread is a path of execution within a process. A process can contain multiple threads.
  + The primary difference is that threads within the same process run in a shared memory space, while processes run in separate memory spaces.  
    Threads are not independent of one another like processes are, and as a result threads share with other threads their code section
  + A thread is the smallest unit of processing that can be performed in an OS.
  + Threads are a popular way to improve the performance of an application through parallelism.
  + **Its consists of:**
    1. Program counter
    2. Stack
    3. Set of registers
    4. A unique id
  + **Advantages:**
    1. Responsiveness: If the process is divided into multiple threads, if one thread completes its execution, then its output can be immediately returned.
    2. Faster context switch: Context switch time between threads is lower compared to process context switch
    3. Resource sharing: Resources like code, data, and files can be shared among all threads within a process.
    4. Communication: Communication between multiple threads is easier, as the threads shares common address space.
    5. Enhanced throughput of the system: If a process is divided into multiple threads, and each thread function is considered as one job, then the number of jobs completed per unit of time is increased, thus increasing the throughput of the system.
* Types of the Threads**:**

1. User level
2. Kernel level

|  |  |  |
| --- | --- | --- |
| **Parameters** | **User Level** | **Kernel Level** |
| Implemented by | Implemented by users | Implemented by os |
| Context Switch time | Context Switch time is less | Context Switch time is more |
| Hardware Support | Not required hardware support | Hardware support is needed |
| Operating system | Any os can user-level threads | Kernel level threads are os specific |
| Example | Java thread | Window solaris |

References:

1. <https://www.youtube.com/watch?v=YO4MGNu2xvI>
2. <https://www.techtarget.com/searchapparchitecture/definition/dependency-injection>
3. <https://www.techopedia.com/definition/27857/thread-operating-systems>