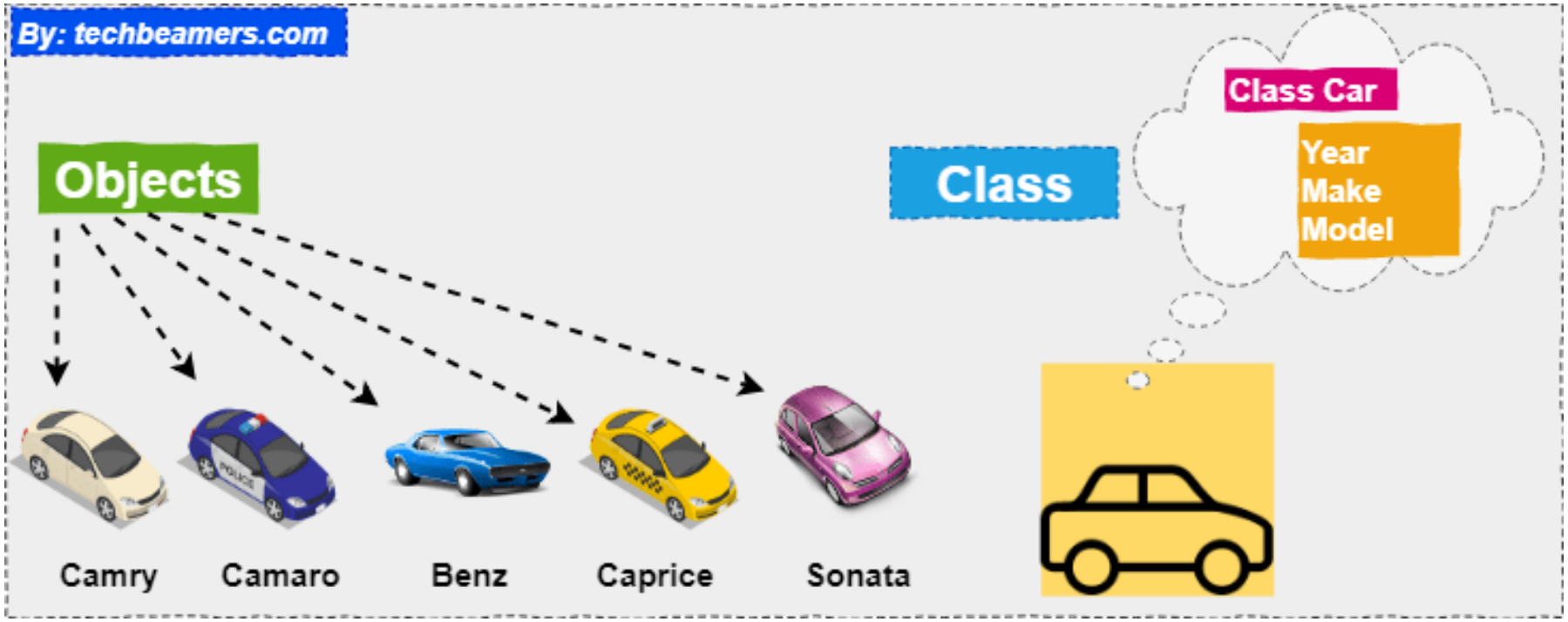
WORKSHOP PEMROGRAMAN JARINGAN
MODUL 6
(MULTIPLEXING SOCKET I/O)

Mochammad Zen Samsono Hadi, ST. MSc. Ph.D

TOPIK PEMBAHASAN

- Python Classes / Objects
- Python Inheritance
- Thread:
  - System fork() => linux
  - Threading => windows

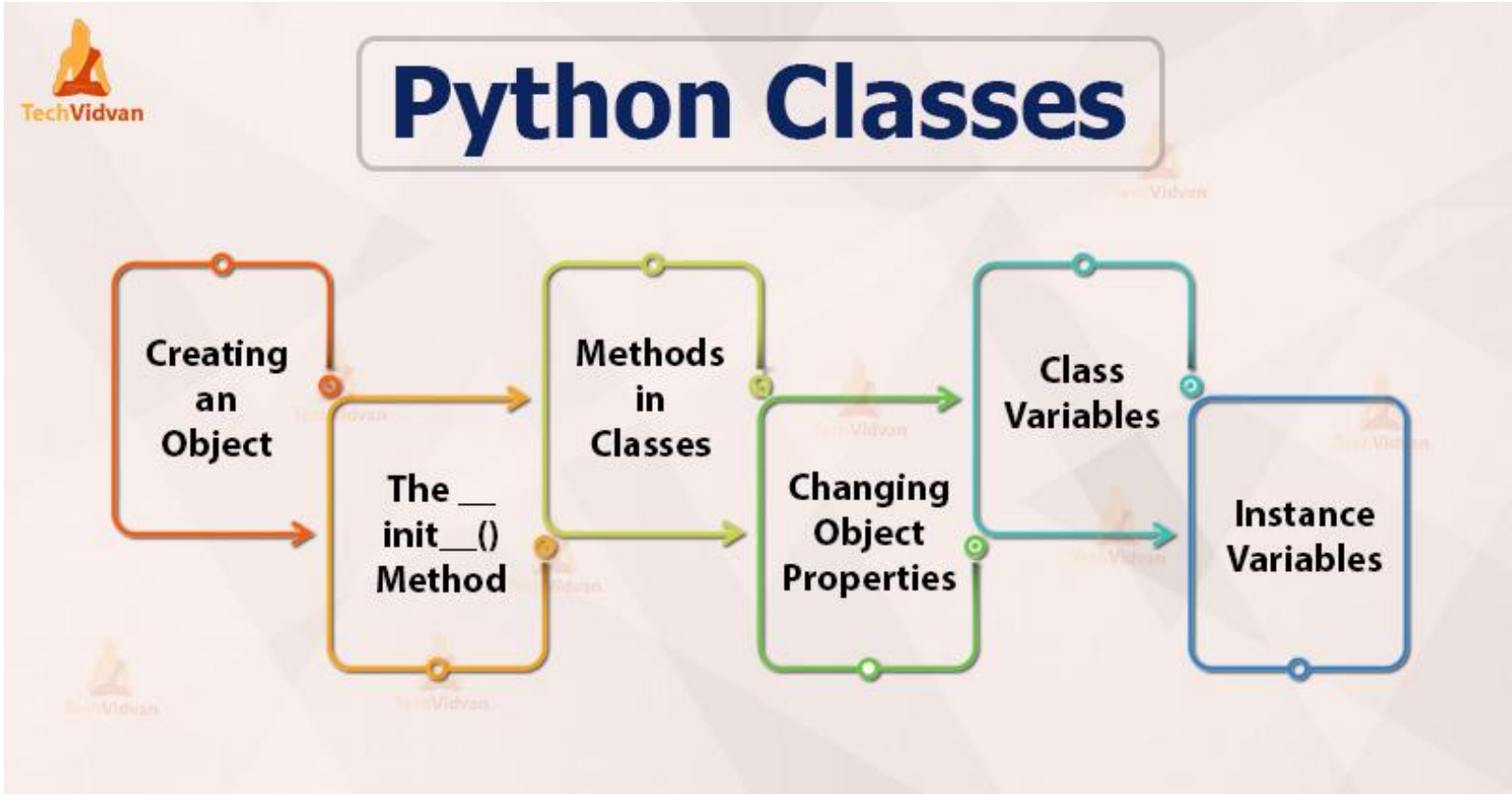
Classes / Objects



- Reference:

https://www.w3schools.com/Python/python_classes.asp

Python Classes / Objects



Membuat Obyek

- Contoh code

```
class MyClass:
  x = 5

p1 = MyClass()
print(p1.x)
```

Output:
5

- Penggunaan __init__() function

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("John", 36)

print(p1.name)
print(p1.age)
```

Output:
John
36

Object Methods

- self digunakan utk mengakses variabel di dalam class

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def myfunc(self):
    print("Hello my name is " + self.name)

p1 = Person("John", 36)
p1.myfunc()
```

Output:
Hello my name is John

Modify Object Properties

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def myfunc(self):
    print("Hello my name is " + self.name)

p1 = Person("John", 36)

p1.age = 40

print(p1.age)
```

Output:
40

Python Inheritance

- Parent class: disebut sebagai base class
- Child class: disebut derived class (inherits dari class lain)
  - pass: mewarisi semua properties dari class

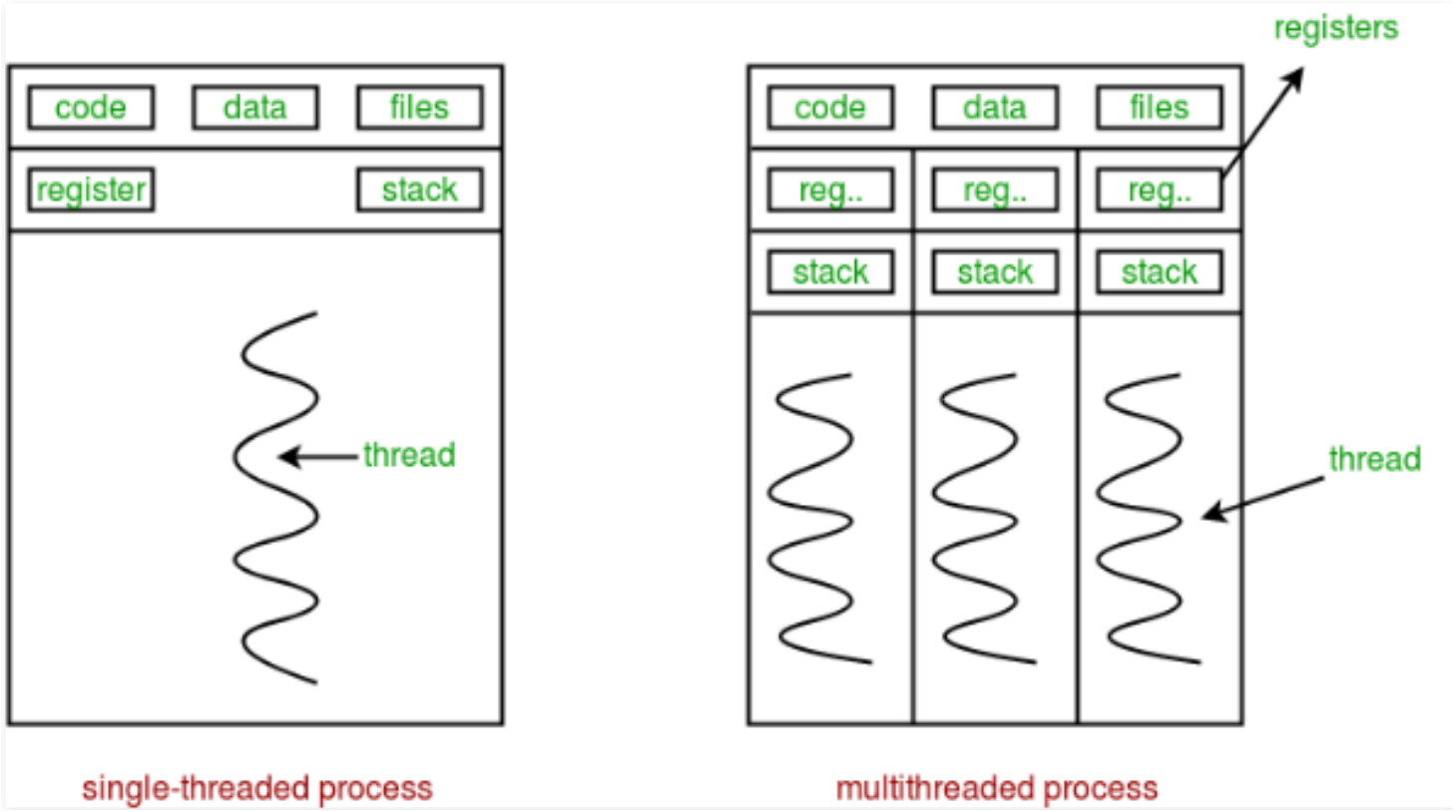
```
class Person:
  def __init__(self, fname, lname):
    self.firstname = fname
    self.lastname = lname

  def printname(self):
    print(self.firstname, self.lastname)
```

```
class Student(Person):
  pass
```

```
x = Person ("John", "Doe")
x.printname()
y = Student("Mike", "Olsen")
y.printname()
```

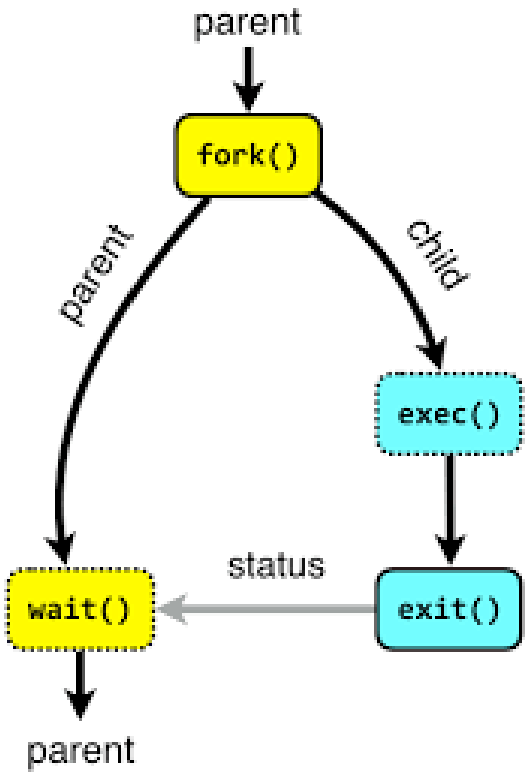
Multithreading

- A **thread** is an entity within a process that can be scheduled for execution. Also, it is the smallest unit of processing that can be performed in an OS (Operating System).



Thread: fork() => Listing 2.1

- System yang berfungsi untuk membuat proses baru



Threading di Python (Listing 2.2)

```
import os
import socket
import threading
import socketserver

SERVER_HOST = 'localhost'
SERVER_PORT = 0 # tells the kernel to pickup a port dynamically
BUF_SIZE = 1024


def client(ip, port, message):
    """ A client to test threading mixin server"""
    # Connect to the server
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect((ip, port))
    try:
        sock.sendall(bytes(message, 'utf-8'))
        response = sock.recv(BUF_SIZE)
        print ("Client received: %s" %response)
    finally:
        sock.close()
```

Threading di Python (Listing 2.2)

```
class ThreadedTCPRequestHandler(socketserver.BaseRequestHandler):
    """ An example of threaded TCP request handler """
    def handle(self):
        data = self.request.recv(1024)
        cur_thread = threading.current_thread()
        response = "%s: %s" %(cur_thread.name, data)
        self.request.sendall(bytes(response, 'utf-8'))

class ThreadedTCPServer(socketserver.ThreadingMixIn,
socketserver.TCPServer):
    """Nothing to add here, inherited everything necessary from parents"""
    pass
```

Untuk menangani koneksi dari client

Threading di Python (Listing 2.2)

```
if __name__ == "__main__":
    # Run server
    server = ThreadedTCPServer((SERVER_HOST, SERVER_PORT),
                               ThreadedTCPRequestHandler)
    ip, port = server.server_address # retrieve ip address

 # Start a thread with the server -- one  thread per request
 server_thread = threading.Thread(target=server.serve_forever)
 # Exit the server thread when the main thread exits
 server_thread.daemon = True
 server_thread.start()
 print ("Server loop running on thread: %s"  %server_thread.name)
 # Run clients
 client(ip, port, "Hello from client 1")
 client(ip, port, "Hello from client 2")
 client(ip, port, "Hello from client 3")
 # Server cleanup
 server.shutdown()
```

```
$ python 2_2_threading_mixin_socket_server.py
Server loop running on thread: Thread-1
Client received: b"Thread-2: b'Hello from client 1'"
Client received: b"Thread-3: b'Hello from client 2'"
Client received: b"Thread-4: b'Hello from client 3'"
```

TUGAS

- Menyusul minggu depan krn ada UTS