# Industrial Software Development (ISDe)

**Exercise 3 (45 min)**

**<u>USE THE OBSERVER DESIGN PATTERN</u>**

**Use abstract classes if needed.**

A **Printer** object has two methods, **process_1( )** and **process_2( )**, which behave differently on working day or on weekend.

---

**working day**:

**process_1( el )** prints the character **el**,
**process_2(el)** prints the UPPERCASE character [ **el.upper( )** ]

**weekend**:
**process_1( el )** prints the UPPERCASE character [ **el.upper( )** ] if **el** is not a number, otherwise it prints the number and the string '**-> IT IS A NUMBER!**'

**process_2(el)** prints the string ' **ASCII ->** ', and the ASCII code [ord(**el**) ]

---

The assigned code [**main.py** and **printer.py**] solves the problem using the STRATEGY DESIGN PATTERN

**New requirements**

Different observers s_1, … , s_n may be interested in a **change of strategy**.
If the strategy changes, the **interested** observers must print

```
‘s_1 ->  New Strategy!’
‘s_2 ->  New Strategy!’
```

If an observer (i.e., s_1) is no more interested in the event, when the strategy changes the output will be

```
‘s_2 ->  New Strategy!’
```

Complete the module **printer.py** and the **main.py**.
The assigned **main.py** contains (as a comment) the correct, expected output.

Be sure to submit both the **main** and the **module**. Your program must work correctly and produce the correct output.