# Practical Learning #5 - 1 Python Exception Handling

#### What are exceptions in Python?

Python has many built-in exceptions which forces your program to output an error when something in it goes wrong.

When these exceptions occur, it causes the current process to stop and passes it to the calling process until it is

handled. If not handled, our program will crash.

- 1. Create a new python file name as PL5-1\_LastnameFirstname.
- 2. Type and try: print(x) #This statement will raise an error, because x is not defined.

### **Catching Exceptions in Python**

In Python, exceptions can be handled using a **try** statement.

A critical operation which can raise exception is placed inside the try clause and the code that handles exception is written in except clause.

3. Modify your code, as shown below.

# The try block will generate an exception, because x is not defined

```
print(x)
except:
  print("An exception occurred")

An exception occurred
An exception occurred
```

## **Catching Specific Exceptions in Python**

You can define as many exception blocks as you want, e.g. if you want to execute a special block of code for a special kind of error

4. Modify your code, as shown below.

```
try:
   print(x)
except NameError:
   print("Variable x is not defined")
except:
   print("Something else went wrong")
```

```
OUTPUT:
Variable x is not defined
```

#### **Else**

You can use the **else** keyword to define a block of code to be executed if no errors were raised:

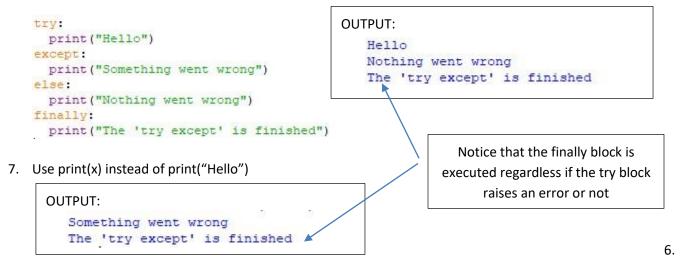
5. Comment your previous code, type the following:

```
try:
   print("Hello")
except:
   print("Something went wrong")
else:
   print("Nothing went wrong")
```

```
OUTPUT:
Hello
Nothing went wrong
```

# **Finally**

The **finally** block, if specified, will be executed regardless if the try block raises an error or not.



Modify your code, as shown below:

8. Save and upload your work.