

# Exceptions

- ▶ Exceptions is a widely used for handling errors or exceptional conditions. Almost all modern languages use exceptions in some form.
- ▶ Some examples of exceptional conditions or error are: division by 0, negative number in a factorial calculation, illegal operations (adding an int and a string), reading beyond end of file etc.
- ▶ Exceptions are handled using try-except blocks.

## Try-except block

```
try:
    try_code
except exception_group1 [as variable1]:
    except_code1
[...
except exception_groupN [as variableN]:
    except_codeN]
[else:
    else_code]
[finally:
    finally_code]
```

The code in [ ] is optional. At least one except clause must be present. The exception\_group can be one exception or a tuple of exceptions. variable if present is bound to the exception that occurred. The else block, if present, is executed if the try\_code executes normally (when no exception). The finally block, if present, is always executed.

## Exception hierarchy

