ex1:

```
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
except:
    print("enter integers only")
print("bye")
```

output1: without any exception

```
hai
enter x value: 3
enter y value: 2
1.5
bye
```

output2: with exception can be handled

```
hai
enter x value: 3
enter y value: 2.3
enter integers only
bye
```

output3: with exception can be handled

```
hai
enter x value: 3
enter y value: 0
enter integers only
bye
```

ex2:

```
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
except ValueError:
    print("enter integers only")
print("bye")
```

output: without any exception

```
hai
enter x value: 3
enter y value: 2
1.5
bye

output2: with exception can be handled
--------
hai
enter x value: 3
enter y value: 2.3
enter integers only
bye

output3: with exception can't be handled
--------
hai
enter x value: 3
enter y value: 0
Traceback (most recent call last):
  File "C:/Python310/e.py", line 5, in <module>
    z=x/y
ZeroDivisionError: division by zero

ex3:
----
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
except Exception as e:
    print(e)
print("bye")

output1: without any exception
--------
hai
enter x value: 3
enter y value: 2
1.5
bye

output2: with exception can be handled
--------
hai
enter x value: 3
enter y value: 2.3
invalid literal for int() with base 10: '2.3'
```

```
bye

output3: with exception can be handled
--------
hai
enter x value: 3
enter y value: 0
division by zero
bye


single try with multiple except blocks:
---------------------------------------
        we can define single try with multiple except blocks,in that case we can
define the default except block must be last.

        syntax
        ------
        try:
           stmt_1
           stmt_2
           .....
           stmt_n
        except ExceptionClassname:
           stmt_1
           stmt_2
           ......
           stmt_n
        except ExceptionClassname:
           stmt_1
           stmt_2
           ......
           stmt_n
        ------
        ------
        except:
           stmt_1
           stmt_2
           ......
           stmt_n

ex:
---
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
    a=[10,20,30,40,50]
```

```
    i=int(input("enter index value: "))
    print(a[i])
    b={'name':'siva','age':29,'sal':2000}
    j=input("enter key: ")
    print(b[j])
    print(c)
except ValueError:
    print("please enter integers only")
except ZeroDivisionError:
    print("sno can't be zero")
except KeyError:
    print("missing key")
except IndexError:
    print("Index Out Of Range")
except:
    print("c is not defined")
print("bye")
```

output1:

```
--------
hai
enter x value: 3
enter y value: 2
1.5
enter index value: 1
20
enter key: age
29
c is not defined
bye
```

output2:

```
--------
hai
enter x value: 3
enter y value: 2
1.5
enter index value: 2
30
enter key: gender
missing key
bye
```

output3:

```
--------
hai
enter x value: 3
enter y value: 2
1.5
enter index value: 5
```

```
Index Out Of Range
bye
```

output4:

```
--------
hai
enter x value: 3
enter y value: 2.3
please enter integers only
bye
```

output5:

```
--------
hai
enter x value: 3
enter y value: 0
sno can't be zero
bye
```

ex2:

```
----
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
    a=[10,20,30,40,50]
    i=int(input("enter index value: "))
    print(a[i])
    b={'name':'siva','age':29,'sal':2000}
    j=input("enter key: ")
    print(b[j])
    print(c)
except Exception as e:
    print(e)
print("bye")
```

output1

```
-------
hai
enter x value: 3
enter y value: 2
1.5
enter index value: 2
30
enter key: age
29
name 'c' is not defined
bye
```

```
output2
-------
hai
enter x value: 3
enter y value: 2
1.5
enter index value: 1
20
enter key: gender
'gender'
bye

output3
-------
hai
enter x value: 3
enter y value: 2
1.5
enter index value: 5
list index out of range
bye

output4
-------
hai
enter x value: 3
enter y value: 2.3
invalid literal for int() with base 10: '2.3'
bye

output5
-------
hai
enter x value: 3
enter y value: 0
division by zero
bye

nested try:
-----------
        we can define a try block inside another try block,is known as a nested
try.

        try: #outer try
           stmt_1
           stmt_2
           .....
           stmt_n
           try: #inner try
```

```
            stmt_1
            stmt_2
            ......
            stmt_n
```

if any exception occured in our outer-try block,that can be handled by using outer-try-except block only,wheather it is not handled our program execution will be terminated abnormally.

ex1:

```
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    try:
        z=x/y
        print(z)
    except ZeroDivisionError:
        print("sno can't be zero")
except ValueError:
    print("please enter integers only")
print("bye")
```

output: without exception

```
hai
enter x value: 3
enter y value: 2
1.5
bye
```

output: with exception can be handled

```
hai
enter x value: 3
enter y value: 2.3
please enter integers only
bye
```

ex2:

```
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    try:
        z=x/y
        print(z)
```

```
    except ValueError:
        print("please enter integers only")
except ZeroDivisionError:
    print("sno can't be zero")

print("bye")
```

output1: with out exception

```
-------
hai
enter x value: 3
enter y value: 2
1.5
bye
```

output2: with exception can't be handled

```
--------
hai
enter x value: 3
enter y value: 2.3
Traceback (most recent call last):
  File "C:/Python310/e.py", line 4, in <module>
    y=int(input("enter y value: "))
ValueError: invalid literal for int() with base 10: '2.3'
```

if any exception occured in our inner-try block,that can be handled by using inner-try-except block,wheather it is not handled that can be handled by using outer-try-except block also.

ex1:

```
----
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    try:
        z=x/y
        print(z)
    except ZeroDivisionError:
        print("sno can't be zero")
except ValueError:
    print("please enter integers only")
print("bye")
```

output: without any exception

```
------
hai
enter x value: 3
enter y value: 2
```

```
1.5
bye

output2: with exception can be handled
-------
hai
enter x value: 3
enter y value: 0
sno can't be zero
bye

ex2:
----
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    try:
        z=x/y
        print(z)
    except ValueError:
        print("please enter integers only")
except ZeroDivisionError:
    print("sno can't be zero")
print("bye")

output: without any exception
------
hai
enter x value: 3
enter y value: 2
1.5
bye

output2: with exception can be handled
-------
hai
enter x value: 3
enter y value: 0
sno can't be zero
bye

ex3:
----
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    try:
        z=x/y
```

```
        print(z)
    except ValueError:
        print("please enter integers only")
except NameError:
    print("sno can't be zero")
print("bye")

output: without any exception
------
hai
enter x value: 3
enter y value: 2
1.5
bye

output2: with exception can't be handled
-------
hai
enter x value: 3
enter y value: 0

ZeroDivisionError: division by zero

finally block
------- ------
        any block which is preceeded by finally keyword,that block is called
finally block.

        finally:
             stmt_1
             stmt_2
             ......
             stmt_n

        the statements must be executed wheather exception is raised or not
raised,even though exception is raised wheather it is handled or not handled,that
type of statements we are represented in finally block.

        in generally,the resource releaseing statements we are represented in
finally block.

        the resource releaseing statements like,
                                File-closeing
                                Database connection closeing
                                Cloud Connection closeing
                                ......
                                ......
ex:
---
print("hai")
```

```
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
    print(z)
except ZeroDivisionError:
    print("sno can't be zero")
finally:
    print("i am in finally block")
print("bye")
```

output1: without exception

```
--------
hai
enter x value: 3
enter y value: 2
1.5
i am in finally block
bye
```

output2: with exception can be handled

```
--------
hai
enter x value: 3
enter y value: 0
sno can't be zero
i am in finally block
bye
```

output3: with exception can't be handeled

```
--------
hai
enter x value: 3
enter y value: 2.3
i am in finally block
ValueError: invalid literal for int() with base 10: '2.3'
```

try else

```
---------
        try:
           stmt_1
           stmt_2
           ......
           stmt_n
        except:
           stmt_1
           stmt_2
           ......
           stmt_n
        else:
```

```
            stmt_1
            stmt_2
            ......
            stmt_n
```

if any exception occured in our try block,then immediately control will goto except block otherwise(without any exception) the control will goto else block and else block is executed.

ex:

```
---
print("hai")
try:
    x=int(input("enter x value: "))
    y=int(input("enter y value: "))
    z=x/y
except ZeroDivisionError:
    print("sno can't be zero")
else:
    print(z)
finally:
    print("i am in finally block")
print("bye")
```

output1: without any exception

```
-------
hai
enter x value: 3
enter y value: 2
1.5
i am in finally block
bye
```

output2: with exception

```
--------
hai
enter x value: 3
enter y value: 0
sno can't be zero
i am in finally block
bye
```