

## Assignment - 27 Exception Handling

1. Write a python script to create a ArithmeticError.

```
print("enter the two numbers")
num1=int(input())
num2=0
res=num1/num2
print(res)
#zero division error sub class of arithmetic error
```

2. Write a python script to create a ValueError.

```
num1=int(input("enter 1st num "))
num2=int(input("enter 2nd num "))
res=num1/num2
print(res)
```

3. Write a python script to handle the ArithmeticError

```
print("enter the numbers")
num1=int(input())
num2=0
try:
    res=num1/num2
except ArithmeticError:
    print("zero division error")
else:
    print(res)
```

4. Write a python script to handle a ValueError.

```
try:
    num1=int(input("enter 1st num "))
    num2=int(input("enter 2nd num "))
    res=num1/num2
except ValueError:
    print("you enter wrong value")
else:
    print(res)
```

5. Write a python script to handle multiple Exception in one try.

```
n1=int(input("enter the value "))
try:
    #n2=input()
    #n2=iint(input())
    n2=int(input("enter the value "))
    if n1>n2:
        print(n1)
```

```

else:
    print(n2)
except TypeError:
    print("typing error")
except ValueError:
    print("you entered wrong value")
except NameError:
    print("you wrote wrong syntax")

```

6. Write a python script to create a calculator with 4 basic operations, and handle a maximum number of exceptions.

```

try:
    print("enter two number ")
    n1,n2=int(input()),int(input())
    if n1==n2:
        raise ArithmeticError
    if n1>n2:
        pass
    else:
        n1,n2=n2,n1
    print("addition is",n1+n2)
    print("multiply is",n1*n2)
    print("subtraction is",n1-n2)
    print("com_division is",n1/n2)
    print("floor_div",n1//n2)
except ValueError:
    print("enter wrong value ")
except ZeroDivisionError:
    print("can not divide by zero")
except ArithmeticError:
    print("value should not same")

```

7. Write a python script to add a finally block for the above script.

```

try:
    print("  start calculator\n")
    print("enter two number ")
    n1,n2=int(input()),int(input())
    if n1==n2:
        raise ArithmeticError
    if n1>n2:
        pass
    else:
        n1,n2=n2,n1
    print("addition is",n1+n2)
    print("multiply is",n1*n2)
    print("subtraction is",n1-n2)
    print("com_division is",n1/n2)
    print("floor_div",n1//n2)
except ValueError:
    print("enter wrong value ")

```

```

except ZeroDivisionError:
    print("can not divide by zero")
except ArithmeticError:
    print("value should not same")
finally:
    print("\n    close calculator")
    print()

```

8. Write a python script to implement try except and else block for division.

```

n1=int(input("enter the number "))
try:
    n2=int(input("enter the number "))
except ValueError:
    print("value is not int type")
else:
    print("complete division",n1/n2)
    print("floor division",n1//n2)

```

9. Write a python script to raise a ValueError.

```

n1=int(input("enter the value "))
try:
    n2=int(input())
    if n2==type(str):
        raise ValueError
except ValueError:
    print("value is not int type")
else:
    print(n1+n2)

```

10. Write a python script to implemented a nested Try Except block.

```

dict = {
    "a": 5,
    "b": 25,
    "c": 125
}

try:
    print(dict["e"])
except KeyError:
    try:
        print("c:", a["c"])
    except:
        print("No value exists for keys 'a' and 'd'")
    finally:
        print("Nested finally")
finally:

```

```
print("Finally")
```