

```
"""
Write an application named GoTooFar in which you declare an array of five
integers and store five values in the array.
Write a try block in which you loop to display each successive element of
the array,
    increasing a subscript by 1 on each pass through the loop. Create a catch
block that catches the exception IndexError and displays the message,
    "Now you've gone too far."
"""
```

```
"""
import numpy as np
arr=np.array([2,3,4,5,6])
try:
    for i in range(int(input("number"))):
        print(arr[i])
except Exception as ob:
    print("Now you've gone too far")
"""
```

```
"""
The int() method requires a String argument, but fails if the String cannot
be converted to an integer.
Write an application in which you try to parse a string that does not
represent an integer value.
Catch the ValueError that is thrown, and then display an appropriate error
message.
"""
```

```
try:
    var=int(input("Enter number"))
    print(var)
except ValueError as ob:
    print(ob)
```

```
"""
Write an application that throws and catches an exception when you attempt
to take the square root of a negative value.
Prompt the user for an input value and try the math.sqrt() method on it.
The application either displays the square root or catches the thrown
Exception and displays an appropriate message.
"""
```

```
"""
import math
try:
    var=math.sqrt(eval(input("Enter a number")))
except Exception as ob:
    print(ob)
else:
    print(var)
```

```
"""
Write an application that displays a series of at least five student ID
numbers (that you have stored in an dictionary)
    and asks the user to enter a numeric test score for the student.
Create a ScoreException class, and raise a ScoreException for the class if
the user does not enter a valid score (less than or equal to 100).
Catch the ScoreException and then display an appropriate message.
In addition, store at 0 for the student's score. At the end of the
```

application, display all the student IDs and scores.

"""

```
import pandas as pd
thisdict={1:"",2:"",3:"",4:"",5:""}
for i in thisdict.keys():
    num=int(input("Enter a Number"))
    try:
        if 0<=num<=100:
            thisdict[i]=num
        else:
            thisdict[i]=0
            raise ValueError("User doesnot enter valid input")
    except ValueError as ob:
        print(ob)
print(pd.Series(thisdict))
```

"""

WAP which has an collection name Dish should comprise dishId, dishName, dishPrice & dishMakeTime as keys .
Write a function which prompts the user to enter the dishId of Dish which he/she wants to purchase.
If the dish with dishId entered by the user is available in the Dish in collection,
then it returns that Dish details else display an exception message dish not found please enter available dish id.
For Exception handling Create a User defined Exception class named as DishNotFoundException and throw an DishNotFountException when dish not found.

"""

```
thisdict={"dishId":["0001","0002","0003","0004","0005"],"dishName":["Poha","Idli","Dosa","Paratha","Samosa"],"dishPrice":[20,45,50,30,15],"dishTime":["20Min","30Min","35Min","20Min","5Min"]}
dish=input("Enter Dish Name")
```

```
try:
    if dish in thisdict["dishName"]:
        print(thisdict["dishId"][thisdict["dishName"].count(dish)])
    else:
        raise ValueError("dish not found")
except ValueError as ob:
    print(ob)
import pandas as pd
data=pd.DataFrame(thisdict)
print(data)

if dish in data["dishName"]:
    print(dish)
else:
    raise ValueError("dish not found")
```