E. Create a class E by inheriting class A and B. Define the constructor so that if no argument is passed while creating an object it will display “Class E object created”, if two (2) arguments are passed it will display “Class E object created with value 1 and value 2”. Also define a method show (), which will show “Showing Class E” or “Showing Class E with value 1 and value 2” if no or two arguments are passed to it.

```
class A:
    def __init__(self):
        print("Class A object created")

    def show(self):
        print("Showing Class A")

class B:
    def __init__(self,val1,val2):
        self.x = val1
        self.y = val2
        print(f"Class B object created with {self.x} \
              and {self.y}")

    def show(self):
        print(f"Showing Class B with {self.x} \
              and {self.y}")

class E(A,B):
    def __init__(self,val1=None,val2=None):
        if val1 is None and val2 is None:
            print("Class E object created")
        if val1 is not None and val2 is not None:
            self.x = val1
            self.y = val2
            print(f"Class E object created with {self.x}\
                   and {self.y}")

    def show(self):
        try:
               print("Showing Class E with", self.x, \
                " and ", self.y)

        except AttributeError:
            print("Showing Class E")
e1 = E()
e2 = E(500,600)

e1.show()
e2.show()
```

Assignment: - 12/ Data Visualization using Python

A. Write a Python program to read the sales data from a CSV file (sample.csv) and plot the region-wise total sales in bar graph and pie chart in single figure.

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

data = pd.read_csv('sample.csv')
regions = list(np.unique(data['Region']))
# # print(data.columns)
sales = []
for r in regions:
    data1 = data.loc[data['Region']==r,['Sales']]
    sales.append(data1['Sales'].sum())

region_wise_sales = {'Regions': regions, 'Total Sales': sales}
region_sale = pd.DataFrame(region_wise_sales)
# # print(region_sale)
figure, (axis1,axis2) = plt.subplots(1, 2)
# print(axis)
axis1.bar(region_sale["Regions"], region_sale["Total Sales"],)
axis2.pie(region_sale["Total Sales"],labels=region_sale["Regions"])
# # axis2.plt.show()
plt.show()
```