1(a) – eval

empno =eval(input("Enter Employee Number:"))  
empname = input("Enter Employee Name:")  
basic = int(input("Enter Basic Pay:"))  
da = basic\*0.5  
hra = basic+0.4  
pf = basic+0.12  
gross = basic + da + hra  
netpay = gross - pf  
print(f"|Employee Pay Slip|\n"  
 f"Employee. No: {empno}\n"  
 f"Employee Name: {empname}\n"  
 f"Basic Pay: {basic}\n"  
 f"Gross Salary: {gross}\n"  
 f"Net Salary: {netpay}")

1(b) – input

distance = float(input("Enter Distance in Metres (m):"))  
time = float(input("Enter time in (s):"))  
speed = distance/time  
print(f"The speed of the car is: {speed} m\\s")

2. Built in

print("Built In Functions")  
x = "heLen OF TrOy"  
y = [1, 10, 8, 4, 3,]  
print(f"Original Test Values:\n 1.{x} \n 2.{y}")  
  
print("\n1. lower() Function:")  
a = x.lower()  
print(a)  
  
print("\n2. upper() Function:")  
a = x.upper()  
print(a)  
  
print("\n3. capitalize() Function:")  
a = x.capitalize()  
print(a)  
  
print("\n4. reverse() Function:")  
y.reverse()  
print(y)  
  
print("\n5. Clear() Function:")  
y.clear()  
print(y)

3. String slicing

print("String slicing:")  
arr = "chocolate"  
print("Original string :", arr)  
print("1. [0:3]",arr[0:3])  
print("2. [0:9:2]", arr[0:9:2])  
print("3. [-1:-8:-1]", arr[-1:-8:-1])

4(a). List Operation

test = [1, "Badassss maa"]  
print("List Functions")  
print("\nOriginal list:", test)  
  
print("\n1. append() Function:")  
test.append("Moye Moye")  
print(test)  
  
print("\n2. reverse() Function:")  
test.reverse()  
print(test)  
  
print("\n 3. sort() Function")  
test.pop()  
print(test)  
  
print("\nList Operation:")  
print("1. Concatenation")  
x = test + test  
print(x)  
  
print("\n2. Multiply")  
y = test \* 3  
print(y)

4(b) List comprehension

print("List Comprehension")  
lst = [1, "x", 4, 6, 90, "apple", 'a', 12]  
print("Original List:", lst)  
result = [lst for lst in lst if type(lst) is int]  
print("After Separating Int values:",result)

5(a) Tuple

rollno = ()  
name = ()  
m1 = ()  
m2 = ()  
m3 = ()  
grade = ()  
rollno = input("Enter Your Roll.No:")  
name = input("Enter Your Name:")  
m1 = input("Enter First Subject Mark:")  
m2 = input("Enter Second Subject Mark:")  
m3 = input("Enter Third Subject Mark:")  
  
total = int(m1 + m2 + m3)  
if total >= 250:  
 grade = 'A'  
elif total >= 200:  
 grade = 'B'  
else:  
 grade ='C'  
print(f'Roll.no:{rollno}\nName:{name}\nGrade:{grade}')

5(b) Dictionary

grades ={"Leo":"A","Rolex":"B", "Vikram":"C", "Dilli":"D","Mark Anotny":"Arrear"}  
keys = grades.keys()  
print("Grades:")  
for i in grades:  
 print(i ,":",grades.get(i))

6 – Operator Overloading

class Overloading:  
 def \_\_init\_\_(self, r, i):  
 self.r = r  
 self.i = i  
 def \_\_add\_\_(self, other):  
 r = self.r + other.r  
 i = self.i + other.i  
 return complex(r, i)  
 def \_\_sub\_\_(self, other):  
 r = self.r - other.r  
 i = self.i - other.i  
 return complex(r,i)  
 def \_\_mul\_\_(self, other):  
 r = self.r \* other.r  
 i = self.i \* other.i  
 return complex(r,i)  
 def \_\_truediv\_\_(self, other):  
 r = self.r/ other.r  
 i = self.i/ other.r  
 return complex(r, i)  
  
  
c1 = Overloading(5, 6)  
c2 = Overloading(4, 7)  
add = c1 + c2  
sub = c1 - c2  
mult = c1 \* c2  
div = c1 / c2  
print("Operator Overloading:")  
print("Original Values:", complex(5, 6), "&", complex(4, 7))  
print(f"\nAddition:{add}\nSubtraction:{sub}")  
print(f"Multiplication:{mult}\nDivision:{div}")

7 – Read CSV

import pandas as pd  
employee = pd.read\_csv("Employee.csv", index\_col=[0])  
print(employee)

8 – Write To Excel

import pandas as pd  
data = [["Troy", "23BDA062", 90, 80, 95, 98, 94, "Pass", "A"],  
 ["Helen", "23BDA068", 80, 98, 78, 86, 99, "Pass", "B"],  
 ["Athena", "23BDA069", 98, 87, 94, 91, 86,"Pass", "A"]]  
col\_names = ["Name", "Roll.no", "English", "French",  
 "C", "FDC", "Maths", "Result", "Grade"]  
df = pd.DataFrame(data,columns=col\_names)  
df.to\_excel("Marksheet.xlsx",index=False)

9 – Combine Excel Files

import pandas as pd  
prod\_col = ["Product ID", "Product Name", "Quantity", "Price"]  
ord\_col = ["Order ID", "Order Name", "Order Date", "Order qty"]  
products = [[1, "Speakers", 5, 30000], [2, "Monitors", 7, 10000]]  
orders = [[1, "Jack Dawson", "30/12/2022", 2], [2, "Rose Dawson", "8/9/2023", 3]]  
pdf = pd.DataFrame(products, columns=prod\_col)  
odf = pd.DataFrame(orders, columns=ord\_col)  
pdf.to\_excel("Product.xlsx", index=False)  
odf.to\_excel("Order.xlsx", index=False)  
df1 = [pdf, odf]  
merged = pd.concat(df1, axis=1)  
merged.to\_excel("Combined.xlsx", index=False)

10 – Preparing Data

import pandas as pd  
date = [["2023-08-01"], ["Jan 1, 2016"], ["2012, May 4"], ["25/12/2017"]]  
df = pd.DataFrame(date, columns=["Date"])  
print(f"Before Formatting:\n{df}")  
df["Date"] = pd.to\_datetime(df["Date"], format="mixed", dayfirst=True)  
df["Date"] = df["Date"].dt.strftime("%d/%m/%Y")  
print(f"\nAfter Formatting to 'DD/MM/YYYY' Format:\n{df}")

11 – Cleaning Data

import pandas as pd  
headers = ["Name", "Customer ID", "Aadhar.No", "Phone Number", "Zipcode"]  
data = [["#Batman", "01-", "59366654@4125", "9568741!124", "641?024"],  
 ["#Batman", "01-", "59366654@4125", "9568741!124", "641?024"],  
 ["$Hulk", "02-", "59003!!94@23146", "98\*\*55741!004", "641?0^55"]]  
df = pd.DataFrame(data, columns=headers)  
print(f"Unclean Dataset:\n{df}")  
df = df.drop\_duplicates()  
col = df.columns  
for i in col:  
 df[i] = df[i].str.replace(r"\W", "", regex=True)  
print(f"\nClean Dataset:\n{df}")

12 – Organizing Data

import pandas as pd  
fields = ["Name", "UG Grade", "PG Grade", "Phd Grade"]  
data = [["Clay Jensen", "Distinction", "First Class", "A"],  
 ["Gal Gadot", "First Class", "Third Class", "B"],  
 ["Zendaya", "Distinction", "Second Class", "C"]]  
df = pd.DataFrame(data, columns=fields)  
print(f"Original Data set:\n\n{df}")  
df["DOB"] = ["01/07/1995", "06/09/2000", "08/03/1999"]  
print(f"\nAfter Adding 'DOB' Column\n\n{df}")  
df = df.drop(["Phd Grade"], axis=1)  
print(f"\nAfter Deleting 'Phd grade' Column:\n\n{df}")