1. Create the following three data-series

```
ds_company = pd.Series(
    ["Google", "Google", "Google", "FB", "FB", "Amazon", "Amazon",
"Amazon", "Amazon"],
    name="Company",
)
ds_employee = pd.Series(
    ["Sam", "Tom", "John", "Jeff", "Cole", "Jason", "Dave", "Bob",
"Tim"],
    name="Employee"
)
ds_salary = pd.Series(
    [125000, 250000, 90000, 18000, 220000, 400000, 350000, 175000,
125000],
    name="Salary"
)
```

2. Create a dataframe using the above three data-series:

```
df = pd.DataFrame(
    {"Company": ds_company, "Employee": ds_employee, "Salary":
ds_salary},
)
df.index = ["a", "b", "c", "d", "e", "f", "g", "h", "i"]
```

3. Assign appropriate labels to the index, as well as to the coulmns of the data-frame.

```
df.index.name = "Index"
df.columns = ["Company", "Employee", "Salary"]
df.columns.name = "Columns"
```

4. Write expression to give raise of 10% to all the employees.

```
df["Salary"] = [(item + .10 * item) for item in df["Salary"]]

# or
df["Salary"] = df["Salary"].apply(lambda salary: int(salary + .10 *
salary))
```

5. Write expresson to give raise of 5% to employees nor working in Google (use np.where)

```
df.Salary = np.where(df["Company"] != "Google", df["Salary"],
df["Salary"] * .05 + df["Salary"])
```

6. Create a new dataframe called ‘df_Google’ which has all employees working in google (use boolean indexing).

```
df_google = df[[True, True, True, False, False, False, False, False,
False]]
# or
df_google = df[np.where(df["Company"] == "Google", True, False)]
```

7. Add a new column to the dataframe called ‘Age’, and initialize it with some integer values.

```
df["Age"] = np.random.randint(10, 80, len(df))
```

8. Use the ‘at[]’ property to locate the age of Employee named ‘Jason’. Update Jason’s age to 40.

```
df.at["f", "Age"] = 40
```

9. Print all the rows from labels ‘b’ to ‘g’ using these two approaches : numeric-indexing, label-indexing.

```
# numeric indexing
print(df[1:7])
# label indexing
print(df["b": "g"])
```

10. Display only the Employee and Salary column information using the following two approaches: (a) using loc (b) using iloc (c) without using loc, iloc

```
# (a) using loc
df_empsal_loc = df.loc[:, ["Employee", "Salary"]]

# (b) using iloc
df_empsal_iloc = df.iloc[:, [1,2]]

# (c) without using loc, iloc
df_empsal = df[["Employee", "Salary"]]
```

11. Print all the employees and their salaries who are working for Amazon. Use the following three approaches : (a) using loc (b) using iloc (c) without using loc, iloc

```
# (a) using loc
df_empsal_loc = df.loc[df["Company"] == "Amazon", ["Employee",
"Salary"]]
print(df_empsal_loc)

# (b) using iloc
df_empsal_iloc = df.iloc[[5,6,7,8], [1,2]]
# or
df_empsal_iloc = df.iloc[list(df.Company == "Amazon"), [1,2]]
print(df_empsal_iloc)
```

```
# (c) without using loc, iloc
df_empsal = df[(df.Company == "Amazon")][["Employee", "Salary"]]
print(df_empsal)
```

12. Drop the Age column from the dataframe.

```
print(df.drop(columns=["Age"]))
```

13. Print all the rows were the salary is greater than 150000.

```
print(df.loc[df.Salary > 150000,  :])
```

14. Create a new dataframe where the labels are ordered in descending order, instead of ascending ordering shown in the original data-frame (use reindexing).

```
new_index = ['i', 'h', 'g', 'f', 'e', 'd', 'c', 'b', 'a']
df.reindex(
      index=new_index,
      columns=["Salary", "Employee", "Company", "Age"]
)
```

15. Construct the following dataframe and then using np.where generate a numpy array such that values between 6 and 8 are repalaced by -1; the rest of the values are copied as it is.

```
df1 = pd.DataFrame(
      np.arange(0, 12).reshape(3, 4),
      columns=["one", "two", "three", "four"]
)

res = np.where((df1 >= 6) & (df1 <= 8), -1, df1)
print(res)
```