```
import pandas as pd

# Load Titanic dataset (mock or real)
data = pd.DataFrame({
    "PassengerId": [1, 2, 3, 4, 5],
    "Name": ["Jack", "Rose", "John", "Lily", "Tom"],
    "Age": [22, None, 35, None, 28],
    "Sex": ["male", "female", "male", "female", "male"],
    "Embarked": ["S", None, "C", "S", None],
    "Cabin": ["C85", None, "E46", None, None]
})

print("Before Cleaning:\n", data.isnull().sum(), "\n")

# Fill missing Age with mean
data["Age"].fillna(data["Age"].mean(), inplace=True)

# Fill missing Embarked with mode
data["Embarked"].fillna(data["Embarked"].mode()[0], inplace=True)

# Drop irrelevant columns
data.drop(columns=["Cabin"], inplace=True)

print("After Cleaning:\n", data.isnull().sum(), "\n")

print("Cleaned Dataset:\n", data)
```

```
Before Cleaning:
 PassengerId    0
Name           0
Age            2
Sex            0
Embarked       2
Cabin          3
dtype: int64

After Cleaning:
 PassengerId    0
Name           0
Age            0
Sex            0
Embarked       0
dtype: int64

Cleaned Dataset:
    PassengerId  Name        Age     Sex Embarked
0            1  Jack  22.000000    male        S
1            2  Rose  28.333333  female        S
2            3  John  35.000000    male        C
3            4  Lily  28.333333  female        S
4            5   Tom  28.000000    male        S
/tmp/ipython-input-513939257.py:16: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on whi

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' o


  data["Age"].fillna(data["Age"].mean(), inplace=True)
/tmp/ipython-input-513939257.py:19: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on whi

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' o


  data["Embarked"].fillna(data["Embarked"].mode()[0], inplace=True)
```

```
import pandas as pd

# Create dataset with inconsistent gender values
students = pd.DataFrame({
    "Name": ["Ravi", "Priya", "John", "Meena", "Amit"],
    "Gender": ["M", "F", "male", "Female", "MALE"]
})

print("Before Cleaning:\n", students, "\n")
```

```
# Standardize gender values
students["Gender"] = students["Gender"].str.lower()
students["Gender"] = students["Gender"].replace({"m": "male", "f": "female"})
students["Gender"] = students["Gender"].replace({"male": "Male", "female": "Female"})

print("After Cleaning:\n", students)
```

```
Before Cleaning:
     Name  Gender
0   Ravi       M
1  Priya       F
2   John    male
3  Meena  Female
4   Amit    MALE

After Cleaning:
     Name  Gender
0   Ravi    Male
1  Priya  Female
2   John    Male
3  Meena  Female
4   Amit    Male
```