**2447147**

**SANDRA BENNY**

**PYTHON PROGRAMMING**

**LAB 4 AND LAB 5**

Lab4.py

from abc import ABC, abstractmethod

class StudioService(ABC):

    @abstractmethod

    def provide\_service(self):

        pass

class RecordingService(StudioService):

    def provide\_service(self):

        return "Congrats! you get to use the recording studio."

class MixingService(StudioService):

    def provide\_service(self):

        return "Congrats! you get use the mixing facilities"

class InstrumentService(StudioService):

    def provide\_service(self):

        return "Congrats! you get to use an instrument."

def main():

    while True:

        print("\nStudio Management System")

        print("1. Recording Service")

        print("2. Mixing Service")

        print("3. Mastering Service")

        print("4. Exit")

        choice = int(input("Enter your choice (1-4): "))

        if choice == 1:

            service = RecordingService()

            print(service.provide\_service())

        elif choice == 2:

            service = MixingService()

            print(service.provide\_service())

        elif choice == 3:

            service = InstrumentService()

            print(service.provide\_service())

        elif choice == 4:

            print("Exiting the Studio Management System. Goodbye!")

            break

        else:

            print("Invalid choice. Please select a number between 1 and 4.")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**OUTPUT:**

Studio Management System

1. Recording Service

2. Mixing Service

3. Mastering Service

4. Exit

Enter your choice (1-4): 1

Congrats! you get to use the recording studio.

Studio Management System

1. Recording Service

2. Mixing Service

3. Mastering Service

4. Exit

Enter your choice (1-4): 2

Congrats! you get use the mixing facilities

Studio Management System

1. Recording Service

2. Mixing Service

3. Mastering Service

4. Exit

Enter your choice (1-4): 3

Congrats! you get to use an instrument.

Studio Management System

1. Recording Service

2. Mixing Service

3. Mastering Service

4. Exit

Enter your choice (1-4): 4

Exiting the Studio Management System. Goodbye!

Lab5.py

class AnalysisError(Exception):

    pass

class TextDataAnalyzer:

    def analyze(self, data):

        try:

            studio\_info = {"title": "Movie Script", "length": "120 pages"}

            director = studio\_info["director"]

            word\_count = len(data.split())

            print(f"The script contains {word\_count} words.")

        except KeyError as e:

            print(f"KeyError: {e}")

        except Exception as e:

            print(f"An error occurred: {e}")

class NumericDataAnalyzer:

    def analyze(self, data):

        try:

            result = data / "string"

            print(f"The result is {result}.")

        except TypeError as e:

            print(f"TypeError: {e}")

        except Exception as e:

            print(f"An error occurred: {e}")

    def analyze\_positive\_number(self, data):

        try:

            if data < 0:

                raise ValueError("Number must be positive")

            print(f"The number {data} is positive.")

        except ValueError as e:

            print(f"ValueError: {e}")

        except Exception as e:

            print(f"An error occurred: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

    text\_analyzer = TextDataAnalyzer()

    numeric\_analyzer = NumericDataAnalyzer()

    sample\_text = "This is a studio script."

    sample\_number = 1000

    negative\_number = -500

    print("\nAnalyzing text data:")

    text\_analyzer.analyze(sample\_text)

    print("\nAnalyzing numeric data:")

    numeric\_analyzer.analyze(sample\_number)

    print("\nAnalyzing a positive number:")

    numeric\_analyzer.analyze\_positive\_number(negative\_number)

**OUTPUT**

Analyzing text data:

KeyError: 'director'

Analyzing numeric data:

TypeError: unsupported operand type(s) for /: 'int' and 'str'

Analyzing a positive number:

ValueError: Number must be positive