S. B. JAIN INSTITUTE OF TECHNOLOGY, MANAGEMENT & RESEARCH, NAGPUR.

Practical No.4

Aim: Write a Java program to create an abstract class Instrument with abstract methods play() and tune(). Create subclasses for Glockenspiel and Violin that extend the Instrument class and implement the respective methods to play and tune each instrument.

Name of Student: Sakar Karnewar
Roll No.: CS23169
Semester/Year: IV / 2nd
Academic Session: 2024-25
Date of Performance:
Date of Submission:

AIM: Write a Java program to create an abstract class Instrument with abstract methods play() and tune(). Create subclasses for Glockenspiel and Violin that extend the Instrument class and implement the respective methods to play and tune each instrument.

OBJECTIVES:

• To understand the concepts of Abstraction. • To implement the program using Abstract class and Abstract method.

THEORY:

ABSTRACTION:

Ways to achieve Abstraction

Abstract Class

Rules for Java Abstract class

Syntax of Abstract class

Abstract Method on java

Example of Abstract Class that has an Abstract Method(Other than Practical)

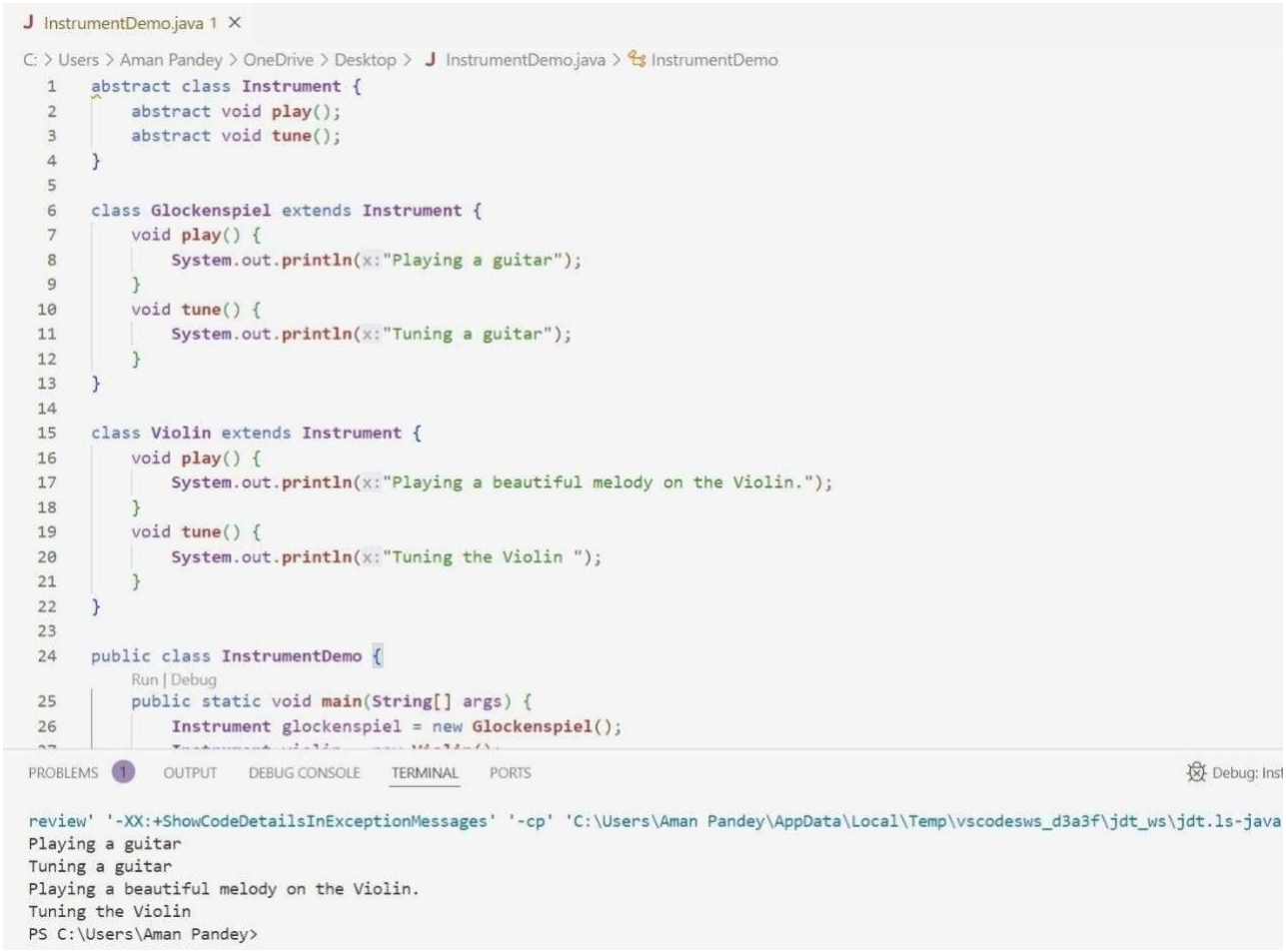
Key Features of Abstract Classes

Observation: Attach Screenshot of Code & output

InstrumentDemo.java 1 ×

C: > Users > Aman Pandey > OneDrive > Desktop > InstrumentDemo.java > InstrumentDemo

```java
abstract class Instrument {
    abstract void play();
    abstract void tune();
}

class Glockenspiel extends Instrument {
    void play() {
        System.out.println(x:"Playing a guitar");
    }
    void tune() {
        System.out.println(x:"Tuning a guitar");
    }
}

class Violin extends Instrument {
    void play() {
        System.out.println(x:"Playing a beautiful melody on the Violin.");
    }
    void tune() {
        System.out.println(x:"Tuning the Violin ");
    }
}

public class InstrumentDemo {
    Run | Debug
    public static void main(String[] args) {
        Instrument glockenspiel = new Glockenspiel();
        Instrument violin = new Violin();

        glockenspiel.play();
        glockenspiel.tune();

        violin.play();
        violin.tune();
    }
}
```



```
J InstrumentDemo.java 1 ×
C: > Users > Aman Pandey > OneDrive > Desktop > J InstrumentDemo.java > InstrumentDemo
1   abstract class Instrument {
2       abstract void play();
3       abstract void tune();
4   }
5
6   class Glockenspiel extends Instrument {
7       void play() {
8           System.out.println(x:"Playing a guitar");
9       }
10      void tune() {
11          System.out.println(x:"Tuning a guitar");
12      }
13  }
14
15  class Violin extends Instrument {
16      void play() {
17          System.out.println(x:"Playing a beautiful melody on the Violin.");
18      }
19      void tune() {
20          System.out.println(x:"Tuning the Violin ");
21      }
22  }
23
24  public class InstrumentDemo {
        Run | Debug
25      public static void main(String[] args) {
26          Instrument glockenspiel = new Glockenspiel();
PROBLEMS 1  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS                Debug: Inst
review' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Aman Pandey\AppData\Local\Temp\vscodesws_d3a3f\jdt_ws\jdt.ls-java
Playing a guitar
Tuning a guitar
Playing a beautiful melody on the Violin.
Tuning the Violin
PS C:\Users\Aman Pandey>
```

INPUT & OUTPUT (With Different Test Cases):

Sr. No.	Input	Output
1		
2		

3		

CONCLUSION:

DISCUSSION AND VIVA VOCE:

1) **When to use Abstract class?**

2) **Can an abstract method be declared with private modifier?**

3) **When to use Abstract method in Java?**

4) **Can we define an abstract method inside non-abstract class (concrete class)?**

5) Why abstract class has constructor even though you cannot create object?

6) Why should we create reference to superclass (abstract class reference)?

REFERENCE:

1. https://www.javatpoint.com/java-program-to-swap-two-string-variables-without-
2. https://www.w3schools.com/java/ref_string_indexof.asp
3. "Programming with Java (6 ed.)", E Balagurusamy,McGraw Hill Education, 2019.