An abstract in simple terms is not concrete. Like A website might be accessed by everyone but depending on role only, we determine the elements or objects a particular user can access. This way of hiding a particular obj or element is called Abstract class , where objects of abstract class are not accessible outside the class. Method with body and without body is possible in abstract class.

An interface in simple term is like a contract between user and provider, where we agree to give one and info and take another, in here user don’t know how it’s done and all backend process , just target info provided, this is also an abstraction, where methods of a certain class can be hidden and exposed.we cannot instantiate an object for interface. Unlike abstract , interface can achieve full abstraction, as in interface ONLY abstract methods are allowed.

Abstract class code

abstract class Music {

abstract void play();

}

class Guitar extends Music {

void play() {

System.out.println("Playing the guitar");

}

}

class Drums extends Music {

void play() {

System.out.println("Playing the drums");

}

}

class Flute extends Music {

void play() {

System.out.println("Playing the flute");

}

}

public class Main {

public static void main(String[] args) {

Guitar guitar = new Guitar();

guitar.play();

Drums drums = new Drums();

drums.play();

Flute flute = new Flute();

flute.play();

}

}

Interface code

interface Music {

void play();

}

class Guitar implements Music {

public void play() {

System.out.println("Playing the guitar");

}

}

class Drums implements Music {

public void play() {

System.out.println("Playing the drums");

}

}

class Flute implements Music {

public void play() {

System.out.println("Playing the flute");

}

}

public class Main {

public static void main(String[] args) {

Guitar guitar = new Guitar();

guitar.play();

Drums drums = new Drums();

drums.play();

Flute flute = new Flute();

flute.play();

}

}