**Activity 5**

In this activity we use the Abstract Class to show implementation of Abstraction.

For this activitiy:

* Create an abstract class *Book* that has:
  + title of type String
  + an abstract method *setTitle()* that takes one String argument.
  + a concrete method *getTitle()* that returns the value of title.
* Create another class that extends the abstract class called *MyBook*.
* *MyBook* defines the *setTitle()* method to assign the value of title as the argument.

In the main method, create an object of MyBook class - newNovel.

Use the setTitle() method to book title to the variable title.

Use the getTitle() method to print the name of the book

--------------------------------------------------------------------------------------------------------------------------------------

abstract class Book {

String title;

abstract void setTitle(String s);

String getTitle() {

return title;

}

}

class MyBook extends Book {

public void setTitle(String s) {

title = s;

}

}

public class Activity5 {

public static void main(String[] args) {

String title = "Skoda Slavia";

Book newNovel = new MyBook();

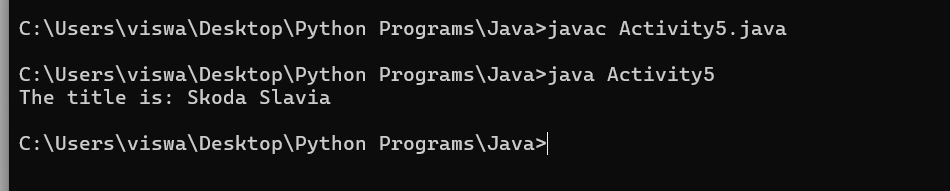
newNovel.setTitle(title);

System.out.println("The title is: " + newNovel.getTitle());

}

}

Output:



Activity6 Program

import java.util.\*;

class Plane {

private List<String> passengers;

private int maxPassengers;

private Date lastTimeTookOf;

private Date lastTimeLanded;

public Plane(int maxPassengers) {

this.maxPassengers = maxPassengers;

this.passengers = new ArrayList<>();

}

public void onboard(String passengerName) {

if(passengers.size() <= maxPassengers) {

this.passengers.add(passengerName);

} else {

System.out.println("Plane is full");

}

}

public Date takeOff() {

this.lastTimeTookOf = new Date();

return lastTimeTookOf;

}

public void land() {

this.lastTimeLanded = new Date();

this.passengers.clear();

}

public Date getLastTimeLanded() {

return lastTimeLanded;

}

public List<String> getPassengers() {

return passengers;

}

}

public class Activity6 {

public static void main(String[] args) throws InterruptedException {

Plane plane = new Plane(10);

plane.onboard("Anand");

plane.onboard("Krishnan");

plane.onboard("Viswanath");

System.out.println("Plane took off at: " + plane.takeOff());

System.out.println("People on the plane: " + plane.getPassengers());

Thread.sleep(5000);

plane.land();

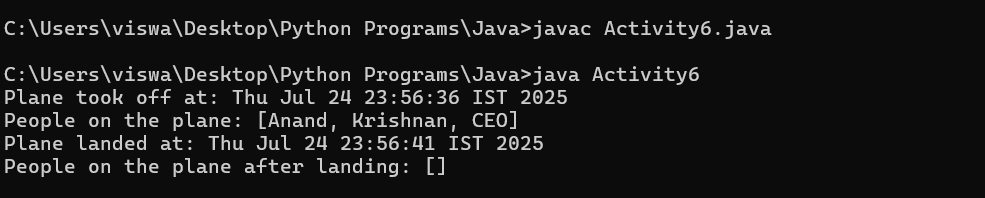
System.out.println("Plane landed at: " + plane.getLastTimeLanded());

System.out.println("People on the plane after landing: " + plane.getPassengers());

}

}

Output:



Activity7

interface BicycleParts {

public int tyres = 2;

public int maxSpeed = 25;

}

interface BicycleOperations {

public void applyBrake(int decrement);

public void speedUp(int increment);

}

class Bicycle implements BicycleParts, BicycleOperations {

public int gears;

public int currentSpeed;

public Bicycle(int gears, int currentSpeed) {

this.gears = gears;

this.currentSpeed = currentSpeed;

}

public void applyBrake(int decrement) {

currentSpeed -= decrement;

System.out.println("Current speed: " + currentSpeed);

}

public void speedUp(int increment) {

currentSpeed += increment;

System.out.println("Current speed: " + currentSpeed);

}

public String bicycleDesc() {

return("No of gears are "+ gears + "\nSpeed of bicycle is " + maxSpeed);

}

}

class MountainBike extends Bicycle {

public int seatHeight;

public MountainBike(int gears, int currentSpeed, int startHeight) {

super(gears, currentSpeed);

seatHeight = startHeight;

}

public void setHeight(int newValue) {

seatHeight = newValue;

}

public String bicycleDesc() {

return (super.bicycleDesc()+ "\nSeat height is " + seatHeight);

}

}

public class Activity7 {

public static void main(String args[]) {

MountainBike mb = new MountainBike(3, 0, 25);

System.out.println(mb.bicycleDesc());

mb.speedUp(20);

mb.applyBrake(5);

}

}

Output:

