

Source Codes:

1st lecture:

```
import java.util.Scanner;

public class Hello {
    static Scanner sc=new Scanner(System.in);

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        System.out.println("May I know your name please?\n");
        String name=sc.nextLine();

        System.out.println("Hello!!"+name);
    }
}
```

2nd lecture:

```
import java.util.Scanner;

public class SecondClass {

    static Scanner myScanner=new Scanner(System.in);

    public static void main(String[] args) {
        int firstIntNumber;                                //declaring variables
        int secondIntNumber;
        char[] cards={'4', '2','3','7','B','K'};
            // Index: 0   1   2   3   4   5
        int[] books=new int[4];

        books[0]=12;
        books[1]=10;
        books[2]=1;
        books[3]=90;

        firstIntNumber=11;                                //this is a starting value
        secondIntNumber=2;

        System.out.println("-----At The Beginning-----");
        printTwoNumbers(firstIntNumber, secondIntNumber);

        float myFloatNumber=10;
        myFloatNumber/=7;

        //secondIntNumber=secondIntNumber+10;              //following line is commented out

        secondIntNumber/=2;
```

```

/*
 * this is a multi line comment
 *
 */

System.out.println("----- After secondIntNumber/=2 -----");
printTwoNumbers(firstIntNumber, secondIntNumber);

secondIntNumber++; //secondIntNumber+=1; secondIntNumber=secondIntNumber+1;

System.out.println("----- After secondIntNumber++ -----");
printTwoNumbers(firstIntNumber, secondIntNumber);

books[3]+=books[1];

System.out.println("----- After books[3]+=books[1] -----");
System.out.println("books[3]= " + books[3] + "\n");

firstIntNumber=(56+34)*(7-9)/4;
System.out.println("----- After firstIntNumber=(56+34)*(7-9)/4 -----");
printTwoNumbers(firstIntNumber, secondIntNumber);

//secondIntNumber--;

System.out.print("Please give us a number to be multiplied by 100: ");
firstIntNumber=myScanner.nextInt();

System.out.println("----- After Getting Input -----");
printTwoNumbers(firstIntNumber, secondIntNumber);

int resultNew=prokriya(firstIntNumber);
int result=firstIntNumber%secondIntNumber;

//-----Printing all the
results-----

System.out.println("firstIntNumber: "+firstIntNumber);
System.out.println("secondIntNumber: "+secondIntNumber);
System.out.println("remainder after firstIntNumber%secondIntNumber: "+result);
System.out.println("myFloatNumber: "+myFloatNumber);
System.out.println("cards[4]: "+cards[4]);
System.out.println("books[3]: "+books[3]);
System.out.println("resultNew: "+resultNew);

}

static public int prokriya(int firstValue) {
    firstValue*=100;
    return firstValue;
}

static public void printTwoNumbers(int firstNumber, int secondNumber){
    System.out.println("First Number: "+firstNumber+"\t SecondNumber:
"+secondNumber);

```

```

        System.out.print("\n");
    }
}

```

3rd lecture:

```

import java.util.Scanner;

public class BookStore {

    static Scanner myScanner=new Scanner(System.in);

    static String[] books={"Java","C","Python"}; //java JAVA JaVa
    static final double studentDiscount=0.3;
    static final double teacherDiscount=0.4;
    static final double alienDiscount=0.0;

    public static void main(String[] args) {
        println("-----WELCOME TO OUR BOOKSTORE-----");
        pr("Which book do you want?\nAns: ");
        String userChoice=myScanner.nextLine();

        if (books[0].toLowerCase().equals(userChoice.toLowerCase())) {
            println("You opted for "+books[0]+" Book.");
            calculatePrice(books[0]);

        }else if (books[1].toLowerCase().equals(userChoice.toLowerCase())) {
            println("You opted for "+books[1]+" Book.");
            calculatePrice(books[1]);

        }else if (books[2].toLowerCase().equals(userChoice.toLowerCase())) {
            println("You opted for "+books[2]+" Book.");
            calculatePrice(books[2]);

        }else {
            println("Sorry, We don't have that book.");
        }

    }

    static void calculatePrice(String bookName){
        pr("Are you a Student,Teacher or alien?\nAns:");
        String answer=myScanner.nextLine();

        double price=200;

        if (answer.toLowerCase().equals("teacher")) {
            price=price-(price*teacherDiscount);
            showPrice(price);
        }else if (answer.toLowerCase().equals("student")) {

```

```

        price=price-(price*studentDiscount);
        showPrice(price);
    }else if (answer.toLowerCase().equals("alien")) {
        price=price-(price*alienDiscount);
        showPrice(price);
    }else {
        println("Sorry, We can't serve you!!");
    }
}

static void showPrice(double price){
    println("Your total payable amount: "+price);
    println("\n-----THANK YOU FOR SHOPPING FROM OUR BOOKSTORE-----");
}

static void println(Object anyObject){
    System.out.println(anyObject);
}

static void pr(Object anyObject){
    System.out.print(anyObject);
}
}

```

4th lecture (rough):

```

import java.util.Iterator;
import java.util.Scanner;

```

```

public class FourthLecture {
    static Scanner myScanner=new Scanner(System.in);

    public static void main(String[] args) {
        int a,b;
        a=10;
        b=20;

        //      int[] numberArray=new int[100];
        //      int i=100;
        //      while (i>0) {
        //          numberArray[i-1]=(i-1)*7;
        //          i--;
        //      }
    }
}

```

```

//      }

int[][] twoDArr=new int[2][4];

twoDArr[0][0]=1;
twoDArr[0][1]=2;
twoDArr[0][2]=4;
twoDArr[0][3]=5;

twoDArr[1][0]=2;
twoDArr[1][1]=4;
twoDArr[1][2]=5;
twoDArr[1][3]=9;

//      println("at 2dArray[1][2]: "+twoDArray[1][2]);
//      1 2 4 5
//      2 4 5 9

//      println("please give your input:");

//taking the input
//      for (int row = 0; row < 2; row++) {
//          for (int column = 0; column < 4; column++) {
//              pr("twoDArray["+row+"]["+column+"] :");
//              twoDArr[row][column]=myScanner.nextInt();
//          }
//          println("\n");
//      }

//now printing the 2darray

int userInput=0;
try {
//      for (int row = 0; row < 2; row++) {
//          for (int column = 0; column < 4; column++) {
//              pr(twoDArr[row][column]+"t");
//          }
//          println("\n");
//      }

```

```

//          }

        pr("Please give a number:");
        userInput=myScanner.nextInt();
        while (userInput<50) {

            prln((100/userInput));
            //pr("Please give a number:");
            //userInput=myScanner.nextInt();
            userInput++;
        }

    } catch (Exception unknownException) {

        unknownException.printStackTrace();

    } finally{
        //userInput;
        prln("came out of try-catch block, with userInput: "+userInput);
    }

    prln("thank you for using our program");

//          for (int j =3; j < 100; j*=2) {
//              prln(j);
//          }

    }

    static int calculateNumber(int number1,int number2){
        number1+=number2;

```

```

        return number1;
    }

    static void prln(Object anyObject){
        System.out.println(anyObject);
    }

    static void pr(Object anyObject){
        System.out.print(anyObject);
    }
}

```

4th lecture (pascal's triangle):

```

import java.util.Scanner;

public class PascalTriangleTwo {

    static Scanner myScanner=new Scanner(System.in);
    public static void main(String[] args) {
        prln("give -1 to exit");
        pr("Give the length of pascal's triangle's length: ");
        int userInput=myScanner.nextInt();
        while (userInput!=-1) {
            if (userInput<-1||userInput>20) {
                prln("please enter something between -1~20");
                userInput=myScanner.nextInt();
                continue;
            }

            printPascalTriangle(userInput);
            prln("\n\n");
            prln("please enter something between -1~20");
            userInput=myScanner.nextInt();
        }
    }
}

```

```

static void printPascalTriangle(int length){
    int[][] pascal=new int[length][length];

    for (int i = 0; i < length; i++) {
        pascal[0][i]=1;
        pascal[i][0]=1;
    }

    for (int row = 1; row < length; row++) {
        for (int column = 1; column < length-row; column++) {

pascal[row][column]=pascal[row][column-1]+pascal[row-1][column];
        }

    }

    for (int row = 0; row < length; row++) {
        for (int column = 0; column < length-row; column++) {
            pr(pascal[row][column]+"t");
        }
        prln("\n");
    }

}

static void prln(Object anyObject){
    System.out.println(anyObject);
}

static void pr(Object anyObject){
    System.out.print(anyObject);
}

}

```

5th Lecture:(LearningClass.java):

```
package FifthLecture;
```

```
public class learningClass {
```

```
    private int aNumber=12;
    private int anotherNumber;
    static private CarClass ourCar=new CarClass();
    static private CarClass ourAnotherCar;
    public static void main(String[] args) {
```

```
        CarClass thirdCar=new CarClass("MAZDA", "Axela", "asdf", 1500, 20);
        SubCarClass ourSubCar=new SubCarClass("BMW", "7 Series", "zzxcz", 3000,
25, 10, "grey");
```

```
        ourAnotherCar=new CarClass("TOYOTA", "Allion", "qwerty", 10000, 30);
```

```
        //first car
        System.out.println("Manufacturer : "+ourAnotherCar.manufacturerName);
        System.out.println("Model Name : "+ourAnotherCar.getModelname());
        System.out.println("Password Match:
"+ourAnotherCar.matchPassword("ahbvjahsb"));
        System.out.println("2nd car CC :"+ourAnotherCar.engineCC);
        System.out.println("Amount of Fuel :
"+ourAnotherCar.presentAmountOfFuel()+"L");
        ourAnotherCar.addFuel(2);
        System.out.println("Amount of Fuel after adding 2L :
"+ourAnotherCar.presentAmountOfFuel()+"L");
```

```
        ourAnotherCar.runForSeconds(60);
        System.out.println("Amount of Fuel after driving for 60s :
"+ourAnotherCar.presentAmountOfFuel()+"L");
```

```
        System.out.println("\n\n");
```

```
        //2nd car
        System.out.println("Manufacturer : "+thirdCar.manufacturerName);
        System.out.println("Model Name : "+thirdCar.getModelname());
        System.out.println("Password Match: "+thirdCar.matchPassword("ahbvjahsb"));
```

```

        System.out.println("2nd car CC :"+thirdCar.engineCC);
        System.out.println("Amount of Fuel : "+thirdCar.presentAmountOfFuel()+"L");
        thirdCar.addFuel(2);
        System.out.println("Amount of Fuel after adding 2L :
"+thirdCar.presentAmountOfFuel()+"L");

        thirdCar.runForSeconds(60);
        System.out.println("Amount of Fuel after driving for 60s :
"+thirdCar.presentAmountOfFuel()+"L");

        //subcar
        System.out.println("\n\n");
        System.out.println("Manufacturer : "+ourSubCar.manufacturerName);
        System.out.println("Model Name : "+ourSubCar.getModelName());
        System.out.println("Password Match:
"+ourSubCar.matchPassword("ahbvjahsb"));
        System.out.println("2nd car CC :"+ourSubCar.engineCC);
        System.out.println("Amount of Fuel : "+ourSubCar.presentAmountOfFuel()+"L");
        ourSubCar.addFuel(2);
        System.out.println("Amount of Fuel after adding 2L :
"+ourSubCar.presentAmountOfFuel()+"L");

        ourSubCar.runForSeconds(60);
        System.out.println("Amount of Fuel after driving for 60s :
"+ourSubCar.presentAmountOfFuel()+"L");
        System.out.println("Charge in Battery :"+ourSubCar.getChargeInBattery());
        System.out.println("Color of car :"+ourSubCar.getColorString());

    }

}

```

5th Lecture:(CarClass.java):

```

package FifthLecture;

public class CarClass {
    public String manufacturerName;
    private String modelName;

```

```

private String carKeyValue;

int engineCC;
private float fuelAmount;
final private float amountOfFuelConsumedPerSecond=0.02f;

public CarClass() {
    manufacturerName=new String();
    modelName=new String();
    engineCC=1000;
    fuelAmount=0;
}
public CarClass(String carName){
    manufacturerName=new String(carName);
    modelName=new String();
    engineCC=0;
    fuelAmount=0;
}

public CarClass(String manu,String model,String key,int engineCC,float fuel) {
    manufacturerName=manu;
    modelName=model;
    carKeyValue=key;
    this.engineCC=engineCC;
    fuelAmount=fuel;
}

public float presentAmountOfFuel() {
    return fuelAmount;
}

public String getManufacturerName() {
    return manufacturerName;
}

public String getModelname() {
    return modelName;
}

public void addFuel(float fuelAmount) {
    this.fuelAmount+=fuelAmount;
}

```

```

        public void runForSeconds(int timeInSecond) {
            float usedFuel=timeInSecond*amountOfFuelConsumedPerSecond;
            fuelAmount-=usedFuel;
        }

        public boolean matchPassword(String userInput) {
            return userInput.equals(carKeyValue);
        }
    }

```

5th Lecture:(SubCarClass.java):

```

package FifthLecture;

public class SubCarClass extends CarClass {

    private float chargeInBattery;
    private String colorString;

    public SubCarClass() {
        // TODO Auto-generated constructor stub
        super();
    }

    public SubCarClass(String manu,String model,String key, int Cc,float fl,float
charge,String col){
        super(manu, model, key, Cc, fl);
        chargeInBattery=charge;
        colorString=col;
    }

    public float getChargeInBattery() {
        return chargeInBattery;
    }
    public void setChargeInBattery(float chargeInBattery) {
        this.chargeInBattery = chargeInBattery;
    }
}

```

```

        public String getColorString() {
            return colorString;
        }
    }
}

```

6th Lecture:(SixthLecture.java):

```

package Sixthlecture;

import FifthLecture.*;
import TestPackage.carClass;

public class SixthLecture {

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        CarClass aCar=new CarClass("MITSUBISHI", "lancer", "poiu", 2500 , 15, 2013);
        SubCarClass aSubCar=new SubCarClass("NISSAN", "350z", "zxcv", 4000, 30,
2011, 12, "Grey");
        AnotherSubCar sixthLectureCar=new AnotherSubCar("PORSCH",
"CareeraGT", "jahdy23gi", 4000, 40, 2012);

        aSubCar.manufacturerName="Mercedes Benz";
        aSubCar.printAllInfo();
        aCar.printAllInfo();
        sixthLectureCar.printAllInfo();
        sixthLectureCar.addAmountOfFuel(12);
        System.out.println("\n\nFuel :"+sixthLectureCar.presentAmountOfFuel());
        System.out.println("MODEL NAME :"+sixthLectureCar.getCarModelName());
        System.out.println("Year :"+sixthLectureCar.getManufacturingYear());

    }

}

```

6th Lecture:(CarInterface.java):

```
package Sixthlecture;

public interface CarInterface {
    public void driveCar(int drivindTimeInSecond);
    public void addAmountOfFuel(int amount) ;
    public int getManufacturingYear();
    public String getCarModelName();
}
```

6th Lecture:(AnotherSubCar.java):

```
package Sixthlecture;

import FifthLecture.CarClass;

public class AnotherSubCar extends CarClass implements CarInterface {
    public AnotherSubCar() {
        super();
    }

    public AnotherSubCar(String carName){
        super(carName);
    }

    public AnotherSubCar(String manu,String model,String key,int engineCC,float fuel) {
        super(manu, model, key, engineCC, fuel);
    }

    public AnotherSubCar(String manu,String model,String key,int engineCC,float fuel,int
year) {
        super(manu, model, key, engineCC, fuel, year);
    }

    @Override
    public void driveCar(int drivindTimeInSecond) {
        // TODO Auto-generated method stub
        runForSeconds(drivindTimeInSecond);
    }
}
```

```

    }

    @Override
    public void addAmountOfFuel(int amount) {
        // TODO Auto-generated method stub
        addFuel(amount);
    }

    @Override
    public int getManufacturingYear() {
        // TODO Auto-generated method stub
        return manufacturingYear;
    }

    @Override
    public String getCarModelName() {
        // TODO Auto-generated method stub
        return getModelname();
    }

}

```

7th Lecture:(Seventh.java):

```

package SeventhLecture;

import java.util.Scanner;

public class Seventh {

    static Scanner myScanner= new Scanner(System.in);
    private static OurThread thread1 = new OurThread("Child 1",0,null,1,500,true);
    private static OurThread thread2 = new OurThread("Child 2",100,thread1,10,1000,true);
    private static OurThread thread3 = new OurThread("Child
3",1000,thread2,100,2000,true);

```

```

public static void main(String[] args) {

    thread1.start();thread2.start();thread3.start();

    System.out.println("Please write a Command: ");

    String commandString=myScanner.nextLine();

    while (!commandString.toLowerCase().equals("exit")) {
        if (commandString.equals("1")) {
            System.out.println("Counter in Thread 1: "+thread1.getCounter());
            threadStatus();
        }else if (commandString.equals("2")) {
            System.out.println("Counter in Thread 2: "+thread2.getCounter());
            threadStatus();
        }else if (commandString.equals("3")) {
            System.out.println("Counter in Thread 3: "+thread3.getCounter());
            threadStatus();
        }else if (commandString.equals("1S")) {
            thread2.stopThread();

            System.out.println("Thread 1 Stopped");
            threadStatus();
        }else if (commandString.equals("2S")) {
            thread2.stopThread();

            System.out.println("Thread 2 Stopped");
            threadStatus();
        }else if (commandString.equals("3S")) {
            thread3.stopThread();

            System.out.println("Thread 3 Stopped");
            threadStatus();
        }else {
            System.out.println("Please give a proper command!!");
        }
    }

    commandString=myScanner.nextLine();

}

```

```

    }

    private static void threadStatus() {
        System.out.println("Thread 1 is alive: "+ thread1.isAlive());
        System.out.println("Thread 2 is alive: "+ thread2.isAlive());
        System.out.println("Thread 3 is alive: "+ thread3.isAlive());
    }
}

```

6th Lecture:(OurThread.java):

```

package SeventhLecture;

public class OurThread extends Thread {

    //-----Variables-----
    private String threadName;
    private int startingPoint;
    private int counter;
    private int adder;
    private int timeDifference;
    private OurThread dependentThread;

    private boolean threadAliveFlag;

    //-----Constructor-----

    public OurThread(String threadName, int startingPoint, OurThread dependentThread,
                     int adder, int timeDifference, boolean threadAliveFlag) {
        super();
        this.threadName = threadName;
        this.startingPoint = startingPoint;
        this.adder = adder;
        this.timeDifference = timeDifference;
    }
}

```

```

        this.threadAliveFlag = threadAliveFlag;

        this.dependentThread=dependentThread;
    }

//-----run() method-----

@Override
public void run() {
    counter=startingPoint;

    if (dependentThread!=null) {
        try {
            dependentThread.join();
        } catch (InterruptedException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }

    while (threadAliveFlag) {
        counter+=adder;

        try {
            sleep(timeDifference);
        } catch (InterruptedException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

}

//-----StopThread method-----

public void stopThread() {

```

```

        threadAliveFlag=false;
    }

//-----Getters and Setters-----
    public String getThreadName() {
        return threadName;
    }

    public void setThreadName(String threadName) {
        this.threadName = threadName;
    }

    public int getStartingPoint() {
        return startingPoint;
    }

    public void setStartingPoint(int startingPoint) {
        this.startingPoint = startingPoint;
    }

    public int getCounter() {
        return counter;
    }

    public void setCounter(int counter) {
        this.counter = counter;
    }

    public int getAdder() {
        return adder;
    }

    public void setAdder(int adder) {
        this.adder = adder;
    }

    public int getTimeDifference() {
        return timeDifference;
    }

    public void setTimeDifference(int timeDifference) {

```

```
        this.timeDifference = timeDifference;
    }

    public boolean getThreadAliveFlag() {
        return threadAliveFlag;
    }

    public void setThreadAliveFlag(boolean threadAliveFlag) {
        this.threadAliveFlag = threadAliveFlag;
    }

}
```