1.

Assignment 1: Write a java program to display “Welcome to Java Programming” and then print your name on a separate line.

**public** **class** one {

**public** **static** **void** main(String args[])

{

System.***out***.println("Welcome to Java Programming");

System.***out***.println("Poonam");

}

}

Assignment 2: Write a Java program to print the result of the following operations. Declare variables and initialize them with given values a. -5 + 8 \* 6 b. (55+9) % 9 c. 20 + -3\*5 / 8 d. 5 + 15 / 3 \* 2 - 8 % 3

**public** **class** one {

**public** **static** **void** main(String args[])

{

**int** a = -5 + 8 \* 6;

**int** b = (55+9) % 9;

**int** c = 20 + -3\*5 / 8;

**int** d = 5 + 15 / 3 \* 2 - 8 % 3;

System.***out***.println("a = "+a);

System.***out***.println("b = "+b);

System.***out***.println("c = "+c);

System.***out***.println("d = "+d);

}

}

Assignment 3: Write a Java program to convert minutes into a number of years and days.

**public** **class** one {

**public** **static** **void** main(String args[])

{

**float** minutes = 35000000;

*getDays*(minutes);

}

**static** **void** getDays(**float** minutes) {

**float** days = minutes / (24 \* 60);

**float** years = minutes / (365\* 24 \* 60) ;

System.***out***.println("Minutes in years = "+years);

System.***out***.println("Minutes in days = "+ days);

}

}

Assignment 4: Write a program to print month in words, based on input month in numbers.(using switch case)

**public** **class** one {

**public** **static** **void** main(String args[])

{

**int** monthNumber = 1;

String monthName;

**switch**(monthNumber) {

**case** 1:

monthName = "January";

**break**;

**case** 2:

monthName = "February";

**break**;

**case** 3:

monthName = "March";

**break**;

**case** 4:

monthName = "April";

**break**;

**case** 5:

monthName = "May";

**break**;

**case** 6:

monthName = "June";

**break**;

**case** 7:

monthName = "July";

**break**;

**case** 8:

monthName = "August";

**break**;

**case** 9:

monthName = "September";

**break**;

**case** 10:

monthName = "October";

**break**;

**case** 11:

monthName = "November";

**break**;

**case** 12:

monthName = "December";

**break**;

**default**:

monthName = "Enter Valid Month Number";

**break**;

}

System.***out***.println(monthName);

}

}

Assignment 5: Write a program that will accept a 4 digit number(assume that the user enters only 4 digit nos.) and print the sum of all the 4 digits. For ex : If the number passed is 3629, the program should print “The sum of all the digits entered is 20”

**import** java.util.Scanner;

**public** **class** one {

**public** **static** **void** main(String args[])

{

Scanner obj = **new** Scanner(System.***in***);

**int** sum = 0;

System.***out***.println("Enter 4 digit number");

**int** inputNumber = obj.nextInt();

**while** (inputNumber != 0) {

sum = sum + inputNumber % 10;

inputNumber = inputNumber / 10;

}

System.***out***.println("The sum of all the digits entered is "+sum);

}

}

Assignment 6: Write a program to find greatest number in an array

**import** java.util.Scanner;

**public** **class** one {

**public** **static** **void** main(String args[])

{

**int** array[] = {122,3,5,6,8,9,6,5,16};

**int** greatest = array[0];

**int** i = 0;

**while**(i<array.length) {

**if**(greatest<array[i]) {

greatest = array[i];

}

i++;

}

System.***out***.println(greatest);

}

}

Assignment 7: Write a Java program to calculate the factorial of a number without using any loop.

**public** **class** one {

**static** **int** *fact* = 1;

**public** **static** **void** main(String args[])

{

System.***out***.println(*factorial*(5));

}

**static** **int** factorial(**int** number) {

**if**(number==0)

**return** number;

**if**(number>1) {

*fact* = number \* *factorial*(number-1);

}

**return** *fact*;

}

}

2.

Assignment 1: Write a program to create a class Book with the following - attributes: -isbn, title, author, price - methods : i. Initialize the data members through parameterized constructor ii. displaydeta ils() to display the details of the book iii. discountedprice() : pass the discount percent, calculate the discount on price and find the amount to be paid after discount - task : Create an object book, initialize the book and display the details along with the discounted price

**public** **class** Book {

**int** isbn;

String title;

String author;

**float** price;

Book(**int** isbn,String title,String author,**float** price){

**this**.isbn = isbn;

**this**.title = title;

**this**.author = author;

**this**.price = price;

}

**void** DisplayDetails()

{

System.***out***.println("isbn: "+ isbn + "title: "+ title + "author: " + author );

}

**void** DiscountedPrice(**int** DiscountPercent)

{

price = price - DiscountPercent\*price/100;

System.***out***.println( "Discounted price" + price);

}

**public** **static** **void** main(String args[]) {

Book ob = **new** Book(10,"Java","Balagurusamy",500);

ob.DisplayDetails();

ob.DiscountedPrice(10);

}

}

Assignment 2: Define a class named Document that contains a member variable of type String named text that stores any textual content for the document. Create a method named toString that returns the text field and also include a method to set this value. Next, define a class for Email that is derived from Document and includes member variables for the sender, recipient, and title of an email message. Implement appropriate accessor and mutator methods. [An accessor is a member function that accesses the contents of an object but does not modify that object; eg: int getX(return x;)A mutator is a member function that can modify an object void setX(int x){this.x=x;} ]The body of the email message should be stored in the inherited variable text. Redefine the toString method to concatenate all text fields.

**class** Document{

String text;

**public** Document() {

}

**public** String toString() {

**return** text;

}

**public** **void** setString(String text) {

**this**.text = text;

}

}

**class** Email **extends** Document{

String sender, recipient, title;

**void** setSender(String sender) {

**this**.sender = sender;

}

String getSender() {

**return** sender;

}

**void** setRecipient(String recipient) {

**this**.recipient = recipient;

}

String setRecipient() {

**return** recipient;

}

**void** setTitle(String title) {

**this**.title = title;

}

String getTitle() {

**return** title;

}

**public** String toString() {

**return** sender + " \n"+ recipient+"\n"+title+"\n"+text;

}

}

**public** **class** Book {

**public** **static** **void** main(String args[]) {

Email ob = **new** Email();

ob.setString("Here is the Email body");

ob.setSender("Sender is Poonam");

ob.setRecipient("Reciepient is Anu");

ob.setTitle("I am the Title");

System.***out***.println(ob.toString());

}

}

Assignment 3: Write a program to create a class Book with the following data members: isbn, title and price. Inherit the class Book to two derived classes : Magazine and Novel with the following data members: Magazine: type Novel : author Populate the details using constructors. Create a magazine and Novel and display the details.

**class** Book{

**int** isbn;

String title;

**float** price;

**public** Book(**int** isbn, String title, **float** price) {

**this**.isbn = isbn;

**this**.title = title;

**this**.price = price;

}

**void** display() {

System.***out***.println("Book : isbn = "+isbn+" title = "+title+" price = "+price);

}

}

**class** Magazine **extends** Book{

String type;

**public** Magazine(**int** isbn, String title, **float** price, String type) {

**super**(isbn, title, price);

**this**.type = type;

}

**void** display() {

System.***out***.println("Magazine : isbn = "+isbn+" title = "+title+" price = "+price+" type = "+type);

}

}

**class** Novel **extends** Book{

String author;

**public** Novel(**int** isbn, String title, **float** price, String author) {

**super**(isbn, title, price);

**this**.author = author;

}

**void** display() {

System.***out***.println("Novel : isbn = "+isbn+" title = "+title+" price = "+price+" author = "+author);

}

}

**public** **class** Caller {

**public** **static** **void** main(String args[]) {

Magazine ob = **new** Magazine(121,"The Title",2550.0f,"Comic");

Novel novelOb = **new** Novel(232,"The Novel",2334.0f,"Drama");

ob.display();

novelOb.display();

}

}

Assignment 4: Define a class named Payment that contains a member variable of type double that stores the amount of the payment and appropriate accessor and mutator methods. Also create a method named paymentDetails that outputs an English sentence to describe the amount of the payment. Next, define a class named CashPayment that is derived from Payment. This class should redefine the paymentDetails method to indicate that the payment is in cash. Include appropriate constructor(s). Define a class named CreditCardPayment that is derived from Payment. This class should contain member variables for the name on the card, expiration date, and credit card number. Include appropriate constructor(s). Finally, redefine the paymentDetails method to include all credit card information in the printout. Create a main method that creates at least two CashPayment and two CreditCardPayment objects with different values and calls paymentDetails for each.

**class** Payment{

**float** payment;

**public** **float** getPayment() {

**return** payment;

}

**public** **void** setPayment(**float** payment) {

**this**.payment = payment;

}

**void** paymentDetails() {

System.***out***.println("Payment = "+payment);

}

}

**class** CashPayment **extends** Payment{

String mode;

**public** CashPayment(String mode) {

**this**.mode = mode;

}

**void** paymentDetails() {

System.***out***.println("Mode is "+ mode);

}

}

**class** CreditCardPayment **extends** Payment {

String name, date;

**long** number;

**public** CreditCardPayment(String name, String date, **long** number) {

**super**();

**this**.name = name;

**this**.date = date;

**this**.number = number;

}

**void** paymentDetails() {

System.***out***.println("Card Info : name = "+name+" date = "+ date+" number = "+number);

}

}

**public** **class** Caller {

**public** **static** **void** main(String args[]) {

CashPayment cashOb1 = **new** CashPayment("Cash");

CashPayment cashOb2 = **new** CashPayment("Card");

CreditCardPayment cardOb1 = **new** CreditCardPayment("ABC","22-12-2222",234545452);

CreditCardPayment cardOb2 = **new** CreditCardPayment("XYZ","2-12-2222",354546565);

cashOb1.paymentDetails();

cardOb1.paymentDetails();

cashOb2.paymentDetails();

cardOb2.paymentDetails();

}

}

Assignment 5: Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute, Guitar. Override the play method inside all three classes printing a message “Piano is playing tan tan tan tan ” for Piano class “Flute is playing toot toot toot toot” for Flute class “Guitar is playing tin tin tin ” for Guitar class You must not allow the user to declare an object of Instrument class. Create an array of 10 Instruments. Assign different type of instrument to Instrument reference. Check for the polymorphic behavior of play method. Use the instanceof operator to print that which object stored at which index of instrument array.

**abstract** **class** Instrument{

**abstract** **void** play();

}

**class** Piano **extends** Instrument{

@Override

**void** play() {

// **TODO** Auto-generated method stub

System.***out***.println("Piano is playing tan tan tan tan");

}

}

**class** Flute **extends** Instrument{

@Override

**void** play() {

// **TODO** Auto-generated method stub

System.***out***.println("Flute is playing toot toot toot toot");

}

}

**class** Guitar **extends** Instrument{

@Override

**void** play() {

// **TODO** Auto-generated method stub

System.***out***.println("Guitar is playing tin tin tin ");

}

}

**public** **class** Caller {

**public** **static** **void** main(String args[]) {

Instrument arr[] = {**new** Guitar(),

**new** Flute(),

**new** Piano(),

**new** Guitar(),

**new** Piano(),

**new** Flute(),

**new** Piano(),

**new** Flute(),

**new** Guitar(),

**new** Piano(),

};

**for**(**int** i = 0; i< arr.length; i++) {

**if**(arr[i] **instanceof** Guitar) {

System.***out***.println(i+" is Guitar");

}**else** **if**(arr[i] **instanceof** Piano) {

System.***out***.println(i+" is Piano");

}**else** **if**(arr[i] **instanceof** Flute) {

System.***out***.println(i+" is Flute");

}

arr[i].play();

}

}

}

Assignment 6: Write an interface called Playable, with a method void play(); Let this interface be placed in a package called music. Write a class called Veena which implements Playable interface. Let this class be placed in a package music.string Write a class called Saxophone which implements Playable interface. Let this class be placed in a package music.wind Sensitivity: Internal & Restricted Write another class Test in a package called live. Then, a. Create an instance of Veena and call play() method b. Create an instance of Saxophone and call play() method c. Place the above instances in a variable of type Playable and then call play()

1.

**package** music;

**public** **interface** Playable {

**void** play();

}

2.

**package** music.string;

**import** music.Playable;

**public** **class** Veena **implements** Playable {

@Override

**public** **void** play() {

// **TODO** Auto-generated method stub

System.***out***.println("Veena Called");

}

}

3.

**package** music.wind;

**import** music.Playable;

**public** **class** Saxophone **implements** Playable {

@Override

**public** **void** play() {

// **TODO** Auto-generated method stub

System.***out***.println("Saxophone Called");

}

}

4.

**package** live;

**import** music.string.Veena;

**import** music.wind.Saxophone;

**import** music.Playable;

**public** **class** Test {

**public** **static** **void** main(String arr[]) {

Veena veena = **new** Veena();

veena.play();

Saxophone saxophone = **new** Saxophone();

saxophone.play();

Playable playable1 = veena;

playable1.play();

Playable playable2 = saxophone;

playable2.play();

}

}

3.

Assignment 1: Write a program to accept name and age of a person from the command prompt(passed as arguments when you execute the class) and ensure that the age entered is >=18 and < 60. Display proper error messages. The program must exit gracefully after displaying the error message in case the arguments passed are not proper. (Hint : Create a user defined exception class for handling errors.)

**class** InvalidAgeException **extends** Exception{

InvalidAgeException(String s){

**super**(s);

}

}

**public** **class** CLA {

**static** **void** validate(**int** age) **throws** InvalidAgeException{

**if**(age<18 || age > 60)

**throw** **new** InvalidAgeException("Age should be between 18 and 60");

**else**

System.***out***.println("welcome to vote");

}

**public** **static** **void** main(String args[]){

**try**{

**int** age = Integer.*parseInt*(args[1]);

*validate*(age);

}**catch**(Exception m){System.***out***.println("Exception occured: "+m);}

System.***out***.println("Name: " + args[0] + " Age: " + args[1]);

}

}

Assignment 2: Write a Program to take care of Number Format Exception if user enters values other that integer for calculating average marks of 2 students. The name of the students and marks in 3 subjects are passed as arguments while executing the program.

**public** **class** CLA {

**public** **static** **void** main(String args[]){

String StudentName1,StudentName2;

**int** sub11, sub21,sub31, sub12,sub22,sub32;

**try** {

StudentName1 = args[0];

StudentName2 = args[4];

sub11 = Integer.*parseInt*(args[1]);

sub21 = Integer.*parseInt*(args[2]);

sub31 = Integer.*parseInt*(args[3]);

sub12 = Integer.*parseInt*(args[5]);

sub22 = Integer.*parseInt*(args[6]);

sub32 = Integer.*parseInt*(args[7]);

System.***out***.println("average of : "+StudentName1+" avg = "+(sub11+sub21+sub31)/3);

System.***out***.println("average of : "+StudentName2+" avg = "+(sub12+sub22+sub32)/3);

}**catch**(NumberFormatException e) {

System.***out***.println("Please enter subject marks in integer");

}

}

}

Assignment 3: Write a program to accept 5 integers passed as arguments while executing the class. Find the average of these 5 nos. Use ArrayIndexOutofBounds exception to handle situation where the user might have entered less than 5 integers.

**public** **class** CLA {

**public** **static** **void** main(String args[]){

**int** num1,num2,num3,num4,num5;

**try** {

num1 = Integer.*parseInt*(args[0]);

num2 = Integer.*parseInt*(args[1]);

num3 = Integer.*parseInt*(args[2]);

num4 = Integer.*parseInt*(args[3]);

num5 = Integer.*parseInt*(args[4]);

System.***out***.println("average is : "+(num1+num2+num3+num4+num5)/3);

}**catch**(ArrayIndexOutOfBoundsException e) {

System.***out***.println("Please enter 5 numbers");

}

}

}

Assignment 4: Write a program to check whether the given string is a palindrome or not. [Hint :You have to extract each character from the beginning and end of the String and compare it with each other. String x=”Malayalam”; char c= x.charAt(i) where i is the index]

**public** **class** palindrome {

**public** **static** **void** main(String args[]) {

String s = "Malayalam";

**int** flag=1;

**int** i = 0, j = s.length() - 1;

**while** (i < j) {

**if** (s.toLowerCase().charAt(i) != s.toLowerCase().charAt(j)) {

flag = 0;

**break**;

}

i++;

j--;

}

**if**(flag==1) {

System.***out***.println("Palindrome");

}

**else** {

System.***out***.println("not palindrome");

}

}

}

Assignment 5: Write a program to check the no.of occurrences of a given character within the given string without using any loop. [Hint: String str=”How was your day today”; char c=’a’; no.of occurrences of a is=3]

**public** **class** Occurrence {

**static** **int** *count* = 0;

**public** **static** **void** main(String args[]) {

String s = "How was your day today";

**char** c = 'a';

*countOccurence*(s, c);

System.***out***.println("Number of occurence of char " + c + " in string " + s + " are " + *count*);

}

**private** **static** **void** countOccurence(String s, **char** c) {

**if** (s.length() > 0) {

**if** (c != s.charAt(s.length() - 1)) {

*countOccurence*(s.substring(0, s.length() - 1), c);

} **else** {

*count*++;

*countOccurence*(s.substring(0, s.length() - 1), c);

}

} **else** {

}

}

}

4.

Assignment 1: Write a Java Program, where one thread prints a number ( Generate a random number using Math.random) and another thread prints the factorial of that given number. Both the outputs should alternate each other. Eg: Number : 2 Factorial of 2 : 2 Number : 5 Factorial of 5 : 120 The program can quit after executing 5 times.

Assignment 2: Write a Java Program which will print the current time on the console every 2 seconds. After doing this activity for 20 seconds the program quits.

**import** java.time.LocalDateTime;

**import** java.time.format.DateTimeFormatter;

**import** java.util.Timer;

**import** java.util.TimerTask;

**public** **class** Caller {

**static** Timer *timer*;

**public** Caller(**int** seconds) {

*timer* = **new** Timer();

*timer*.schedule(**new** RemindTask(),0, seconds\*1000);

}

**class** RemindTask **extends** TimerTask {

**int** numWarningBeeps = 3;

**public** **void** run() {

DateTimeFormatter dtf = DateTimeFormatter.*ofPattern*("yyyy/MM/dd HH:mm:ss");

LocalDateTime now = LocalDateTime.*now*();

System.***out***.println(dtf.format(now));

}

}

**public** **static** **void** main(String args[]) {

**new** Caller(2);

**new** Timer().schedule(**new** TimerTask() {

@Override

**public** **void** run() {

// **TODO** Auto-generated method stub

*timer*.cancel();

System.*exit*(0);

}

}, 20\*1000);

System.***out***.println("Task scheduled.");

}

}

Assignment 3: Create an Employee class with the related attributes and behaviours. Create one more class EmployeeDB which has the following methods. a. boolean addEmployee(Employee e) b. boolean deleteEmployee(int eCode) c. String showPaySlip(int eCode) d. Employee[] listAll() Use an ArrayList which will be used to store the emplyees and use enumeration/iterator to process the employees.

**package** test;

**import** java.util.ArrayList;

**import** java.util.Iterator;

**class** Employee{

String name;

**int** EmpId;

Employee(String name, **int** EmpId){

**this**.name = name;

**this**.EmpId = EmpId;

}

**void** displayDetails() {

System.***out***.println("Employee name:"+ name + "Employee Id" + EmpId);

}

}

**public** **class** EmployeeDB{

**boolean** addEmployee(Employee e) {

**return** **false**;

}

**boolean** deleteEmployee(**int** eCode) {

**return** **false**;

}

String showPaySlip(**int** eCode) {

**return** eCode+"";

}

Employee[] listAll() {

**return** **new** Employee[] {**new** Employee("new",10)};

}

**public** **static** **void** main(String args[]){

//Creating user-defined class objects

Employee s1=**new** Employee("Sonoo",23);

Employee s2=**new** Employee("Ravi",21);

Employee s3=**new** Employee("Hanumat",25);

//creating arraylist

ArrayList<Employee> al=**new** ArrayList<Employee>();

al.add(s1);//adding Student class object

al.add(s2);

al.add(s3);

//Getting Iterator

Iterator itr=al.iterator();

//traversing elements of ArrayList object

**while**(itr.hasNext()){

Employee st=(Employee)itr.next();

System.***out***.println(st.name+" "+st.EmpId);

}

}

}

Assignment 4: Write a program creates a HashMap to store name and phone number (Telephone book). When name is give, we can get back the corresponding phone number.

**import** java.util.HashMap;

**public** **class** Caller {

**public** **static** **void** main(String args[]) {

HashMap telephoneBook = **new** HashMap();

telephoneBook.put("Anu", "9784845656");

telephoneBook.put("Mishu", "85675849");

telephoneBook.put("Priya", "48374586");

telephoneBook.put("Poonam", "129494959");

System.***out***.println(telephoneBook.get("Poonam"));

}

}

Assignment 5: Write a program to store a group of employee names into a HashSet, retrieve the elements one by one using an Iterator.

**import** java.util.HashSet;

**public** **class** Caller {

**public** **static** **void** main(String args[]) {

HashSet hashSet = **new** HashSet();

hashSet.add("Emp1");

hashSet.add("Emp2");

hashSet.add("Emp3");

hashSet.add("Emp4");

hashSet.add("Emp5");

hashSet.add("Emp6");

hashSet.forEach(e ->{System.***out***.println(e);});

}

}

Assignment 6: Develop a java class that has finalize method which displays “Finalize method called”. Create another class which creates objects of the previous class and it uses the same object reference for creating these objects. For example, if A1 is the class name, then the objects are created as Sensitivity: Internal & Restricted below : A1 a = new A1(); a = new A1(); a = new A1(); When the statement Runtime.getRuntime().gc() is invoked, how many times the finalize method is called

**import** java.util.HashSet;

**public** **class** Caller {

**public** **static** **void** main(String args[]) {

AI a = **new** AI();

a = **new** AI();

a = **new** AI();

Runtime.*getRuntime*().gc();

}

}

**class** AI{

@Override

**protected** **void** finalize()

{

System.***out***.println("finalize method called"); // 2 times

}

}

Topic 5: Command Line Args, System Properties, Packaging

Assignment 1: Create a package called test package; Define a class called foundation inside the test package; Inside the class, you need to define 4 integer variables; Var1 as private; Var2 as default; Var3 as protected; Var4 as public; Import this class and packages in another class. Try to access all 4 variables of the foundation class and see what variables are accessible and what are not accessible.

**package** test;

**public** **class** foundation {

**private** **int** Var1 ;

**int** Var2 ;

**protected** **int** Var3 ;

**public** **int** Var4 ;

}

**import** test.foundation;

**public** **class** Access {

**public** **static** **void** main(String args[]) {

foundation ob = **new** foundation();

/\* Var1, Var2, Var3 are not accessible

System.out.println(ob.Var1);

System.out.println(ob.Var2);

System.out.println(ob.Var3); \*/

System.***out***.println(ob.Var4);// Accessible

}

}

Assignment 2. Write a Program to accept two Strings Wipro Bangalore as command line arguments and print the output “Wipro Technologies Bangalore” If the command line is “ABC Mumbai”, then it should print “ABC Technologies Mumbai” .

**public** **class** Cancatenate {

**public** **static** **void** main(String args[]) {

System.***out***.println(args[0] + " Technologies " + args[1]);

}

}

Assignment 3: Create a package called Automobile. Define an abstract class called Vehicle. Vehicle class has the following abstract methods: public String modelName() public String registrationNumber() public String ownerName() Create TwoWheeler subpackage under Automobile package Hero class extends Automobile.vehicle class public int speed() – Returns the current speed of the vehicle. public void radio() – provides facility to control the radio device Honda class extends Automobile.vehicle class public int speed()– Returns the current speed of the vehicle. Sensitivity: Internal & Restricted public int cdplayer() – provides facility to control the cd player device which is available in the car and test all the methods by invoking them.

**package** Automobile;

**public** **abstract** **class** vehicles {

**public** **abstract** String modelName();

**public** **abstract** String registrationNumber();

**public** **abstract** String ownerName();

}

**package** Automobile.TwoWheeler;

**import** Automobile.vehicles;

**public** **class** Hero **extends** vehicles{

@Override

**public** String modelName() {

// **TODO** Auto-generated method stub

**return** "Hero";

}

@Override

**public** String registrationNumber() {

// **TODO** Auto-generated method stub

**return** "12345";

}

@Override

**public** String ownerName() {

// **TODO** Auto-generated method stub

**return** "Owner1";

}

**public** **int** speed() {

**return** 60;

}

**public** **void** radio() {

System.***out***.println("Radio");

}

}

**class** Honda **extends** vehicles{

@Override

**public** String modelName() {

// **TODO** Auto-generated method stub

**return** "Honda";

}

@Override

**public** String registrationNumber() {

// **TODO** Auto-generated method stub

**return** "24354";

}

@Override

**public** String ownerName() {

// **TODO** Auto-generated method stub

**return** "Owner2";

}

**public** **int** speed() {

**return** 65;

}

**public** **int** cdplayer() {

**return** 2;

}

}