|  |
| --- |
| Exp: 09 Date: 04/10/2023  JAVA PROGRAMMING LAB 9 |

Name: VIJAI SURIA M

Reg No.: 2021503568

1. **Write a program to show single inheritance**

[CO2; BL: 3]

Points: 25

Class LivingBeing

Methods:

Breath()

Response()

Class Animal

Methods:

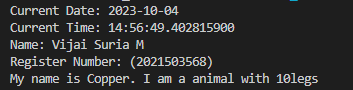
Walk()

NoOfLegs()

**CODE:**

|  |
| --- |
| import java.time.LocalDate;  import java.time.LocalTime;  class LivingBeing {  String name;  LivingBeing(String name) {  this.name = name;  }  protected void breath() {  System.out.println("I am breating........");  }  protected void response() {  System.out.println(  "Hi, My response: My name is " + name + "I am a Living being. \n Thank youu!! \n Hava a Nice day :)");  }  public String toString() {  return "My name is " + name;  }  }  class Animal extends LivingBeing {  int legs;  Animal(int legs, String name) {  super(name);  this.legs = legs;  }  void setlegs(int legs) {  this.legs = legs;  }  void getlegs(int legs) {  System.out.println("No. of legs: " + legs);  }  void walk() {  System.out.println("I am walking......");  }  @Override  public String toString() {  return super.toString() + ". I am a animal with " + legs + "legs";  }  }  public class SingleInheritance3568 {  public static void main(String args[]) {  System.out.println("Current Date: " + LocalDate.now());  System.out.println("Current Time: " + LocalTime.now());  System.out.println("Name: Vijai Suria M \nRegister Number: (2021503568)");  Animal obj = new Animal(10, "Copper");  System.out.println(obj);  }  } |

**OUTPUT**:



1. **Write a program to show Multilevel inheritance**

**[CO2: BL:3] points:25**

**Class LivingBeing**

**Methods:**

**Breath()**

**Response()**

**Class Animal**

**Methods:**

**Walk()**

**NoOfLegs()**

**Class Cat**

**Methods:**

**Meow()**

**Class Dog**

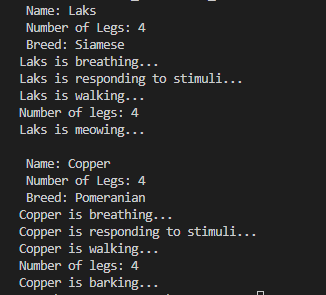
**Methods:**

**Bark()**

**CODE:**

|  |
| --- |
| class LivingBeing {  String name;  LivingBeing(String name) {  this.name = name;  }  void Breath() {  System.out.println(name + " is breathing...");  }  void Response() {  System.out.println(name + " is responding to stimuli...");  }  public String toString() {  return " Name: " + name;  }  }  // First-level derived class  class Animal extends LivingBeing {  int numberOfLegs;  Animal(String name, int numberOfLegs) {  super(name);  this.numberOfLegs = numberOfLegs;  }  void Walk() {  System.out.println(name + " is walking...");  }  int NoOfLegs() {  return numberOfLegs;  }  @Override  public String toString() {  return super.toString() + "\n Number of Legs: " + numberOfLegs;  }  }  // Second-level derived class  class Cat extends Animal {  String breed;  Cat(String name, int numberOfLegs, String breed) {  super(name, numberOfLegs);  this.breed = breed;  }  void Meow() {  System.out.println(name + " is meowing...");  }  @Override  public String toString() {  return super.toString() + "\n Breed: " + breed;  }  }  // Second-level derived class  class Dog extends Animal {  String breed;  Dog(String name, int numberOfLegs, String breed) {  super(name, numberOfLegs);  this.breed = breed;  }  void Bark() {  System.out.println(name + " is barking...");  }  @Override  public String toString() {  return super.toString()  + "\n Breed: " + breed;  }  }  public class MultilevelInheritance3568 {  public static void main(String[] args) {  Cat cat = new Cat("Laks", 4, "Siamese");  Dog dog = new Dog("Copper", 4, "Pomeranian");  System.out.println(cat);  cat.toString();  cat.Breath();  cat.Response();  cat.Walk();  System.out.println("Number of legs: " + cat.NoOfLegs());  cat.Meow();  System.out.println();  System.out.println(dog);  dog.toString();  dog.Breath();  dog.Response();  dog.Walk();  System.out.println("Number of legs: " + dog.NoOfLegs());  dog.Bark();  }  } |

**OUTPUT**:



1. **Write a program in Java to create messaging service like WhatsApp that uses single inheritance, multilevel inheritance, and hierarchical inheritance.**

**[CO:2, BL:6] points:50**

class User:

Attributes

Name

phoneNumber

status

subclass Contact: (Stores contacts)

Attributes

contactList code

SNIPPET:

class Contact extends User {

protected User[] contacts;

protected int contactCount;

public void addContact(User user) {...}

}

Class Message: Contains information about message

Attributes

Sender

Receiver

messageContent

subclass Chat: represent chat between two users.

Attributes

Users (an array of User objects)

messages (an array of Message objects)

Methods

addMessages: add message to the chat displayChatHistory: display the chat history

CODE SNIPPET:

class Chat {

private User[] participants;

private Message[] messages;

}

Create a main class WhatsAppto test the concepts

public class WhatsApp {

public static void main(String[] args) {

User alice = new User("Alice", "+1234567890", "Available");

User bob = new User("Bob", "+9876543210", "Away");

Contact aliceContacts = new Contact("Alice", "+1234567890", "Available", 10);

aliceContacts.addContact(bob);

Message message1 = new Message(alice, bob, "Hi, Bob!");

Message message2 = new Message(bob, alice, "Hello, Alice!");

Chat chat = new Chat(alice, bob, 100);chat.addMessage(message1);

chat.addMessage(message2);chat.displayChatHistory();

}

}

**CODE**:

|  |
| --- |
| import java.time.LocalDate;  import java.time.LocalTime;  class User {  protected String name;  protected String phoneNumber;  protected String status;  // Constructor for User class  public User(String name, String phoneNumber, String status) {  this.name = name;  this.phoneNumber = phoneNumber;  this.status = status;  }  }  class Contact extends User {  protected User[] contacts;  protected int contactCount;  // Constructor for Contact class  public Contact(String name, String phoneNumber, String status, int maxContacts) {  super(name, phoneNumber, status);  this.contacts = new User[maxContacts];  this.contactCount = 0;  }  // Method to add a contact  public void addContact(User user) {  if (contactCount < contacts.length) {  contacts[contactCount] = user;  contactCount++;  System.out.println(name + " added " + user.name + " to contacts.");  } else {  System.out.println("Contact list is full. Cannot add more contacts.");  }  }  }  class Message {  protected User sender;  protected User receiver;  protected String messageContent;  // Constructor for Message class  public Message(User sender, User receiver, String messageContent) {  this.sender = sender;  this.receiver = receiver;  this.messageContent = messageContent;  }  }  class Chat {  private User[] participants;  private Message[] messages;  private int messageCount;  // Constructor for Chat class  public Chat(User user1, User user2, int maxMessages) {  this.participants = new User[]{user1, user2};  this.messages = new Message[maxMessages];  this.messageCount = 0;  }  // Method to add a message to the chat  public void addMessage(Message message) {  if (messageCount < messages.length) {  messages[messageCount] = message;  messageCount++;  } else {  System.out.println("Chat history is full. Cannot add more messages.");  }  }  // Method to display the chat history  public void displayChatHistory() {  System.out.println("Chat History between " + participants[0].name + " and " + participants[1].name);  for (int i = 0; i < messageCount; i++) {  System.out.println(participants[0].name + ": " + messages[i].messageContent);  System.out.println(participants[1].name + ": " + messages[i].messageContent);  }  }  }  public class WhatsApp3568 {  public static void main(String[] args) {  // Displaying current date and time  System.out.println("Current Date: " + LocalDate.now());  System.out.println("Current Time: " + LocalTime.now());  System.out.println("Name: Vijai Suria M \nRegister Number: (2021503568)");  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n");  // Creating User, Contact, and Message objects  User alice = new User("Alice", "+1234567890", "Available");  User bob = new User("Bob", "+9876543210", "Away");  Contact aliceContacts = new Contact("Alice", "+1234567890", "Available", 10);  aliceContacts.addContact(bob);  Message message1 = new Message(alice, bob, "Hi, Bob!");  Message message2 = new Message(bob, alice, "Hello, Alice!");  // Creating a Chat object and adding messages  Chat chat = new Chat(alice, bob, 100);  chat.addMessage(message1);  chat.addMessage(message2);  // Displaying chat history  chat.displayChatHistory();  }  } |

**OUTPUT**:

