FULL STACK DEVELOPMENT – WORKSHEET - C

1. Write a program to implement hybrid and Hierarchical inheritance in Java through an own created example.

```
Ans 1- class Vehicle {
   void display() {
      System.out.println("This is a vehicle.");
   }
}

class Car extends Vehicle {
   void display() {
      System.out.println("This is a car.");
   }
}

class Bike extends Vehicle {
   void display() {
      System.out.println("This is a bike.");
   }
}

class SportsCar extends Car {
   void display() {
      System.out.println("This is a sports car.");
   }
}

class ElectricBike extends Bike {
   void display() {
      System.out.println("This is an electric bike.");
   }
}

public class InheritanceExample {
    public static void main(String[] args) {
        SportsCar sportsCar = new SportsCar();
        ElectricBike electricBike = new ElectricBike();

        sportsCar.display();
        electricBike.display();
    }
}
```

2. Write a small program to implement polymorphism for different shapes (for example, shape consist- square, circle, etc.)

```
Ans 2-    abstract class Shape {
    abstract double area();
}

class Square extends Shape {
    private double side;

    public Square(double side) {
        this.side = side;
    }

    @Override
    double area() {
        return side * side;
    }
}

class Circle extends Shape {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    double area() {
        return Math.PI * radius * radius;
    }
}

public class PolymorphismExample {
    public static void main(String[] args) {
        Shape[] shapes = new Shape[2];
        shapes[0] = new Square(5);
        shapes[1] = new Circle(3);

        for (Shape shape : shapes) {
            System.out.println("Area: " + shape.area());
        }
    }
}
```

3. Write a program to invoke constructor and then utilize to take inputs for rating of any organization.

Ans3-   import java.util.Scanner;

```
class Organization {
    private String name;
    private int rating;
    public Organization(String name) {
        this.name = name;
        this.rating = 0;
    }

    public void setRating(int rating) {
        if (rating >= 1 && rating <= 5) {
            this.rating = rating;
        } else {
            System.out.println("Invalid rating. Please enter a rating between 1 and 5.");
        }
    }

    public int getRating() {
        return rating;
    }
}

public class OrganizationRating {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the name of the organization: ");
        String orgName = scanner.nextLine();

        Organization org = new Organization(orgName);

        while (true) {
            try {
                System.out.print("Enter a rating for the organization (1-5): ");
                int rating = Integer.parseInt(scanner.nextLine());
                org.setRating(rating);

                break;

            } catch (NumberFormatException e) {

                System.out.println("Invalid input. Please enter a valid rating.");

            }

        }


        System.out.println("Thank you! " + orgName + " has been rated with " + org.getRating() + " stars.");


        scanner.close();

    }

}
```

4. Write a program to read text from input stream using Buffer Reader class.

Ans4- import java.io.BufferedReader;

```
import java.io.IOException;

import java.io.InputStreamReader;


public class BufferedReaderExample {

    public static void main(String[] args) {

        BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));


        try {

            System.out.println("Enter text (type 'exit' to quit):");


            String line;

            while ((line = reader.readLine()) != null) {

                if (line.equalsIgnoreCase("exit")) {

                    break;

                }
                System.out.println("You entered: " + line);
            }

            System.out.println("Exiting...");
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                if (reader != null) {
                    reader.close();
                }
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

5. Write a program for creation of single linked list in Java.

```
Ans5-  class Node {
    int data;
    Node next;

    public Node(int data) {
        this.data = data;
        this.next = null;
    }
}

class LinkedList {
    Node head;
    public LinkedList() {
        this.head = null;
    }

    public void insert(int data) {
        Node newNode = new Node(data);
        if (head == null) {
            head = newNode;
        } else {
            Node current = head;
            while (current.next != null) {
                current = current.next;
            }
            current.next = newNode;
        }
    }

    public void display() {
        Node current = head;
        while (current != null) {
            System.out.print(current.data + " -> ");
            current = current.next;
        }
        System.out.println("null");
    }
}

public class Main {
    public static void main(String[] args) {
        LinkedList list = new LinkedList();
        list.insert(1);
        list.insert(2);
        list.insert(3);
        list.insert(4);

        System.out.println("Linked List elements:");
        list.display();
    }
}
```