

A decorative border surrounds the central text, featuring various science and technology icons. At the top, there is a DNA double helix, three test tubes with colored liquids, a satellite dish, and a computer monitor. On the left side, from top to bottom, are a laptop, a test tube, a calculator, a horseshoe magnet, a microscope, a satellite dish, a molecular model, and gears. On the right side, from top to bottom, are a computer monitor, three test tubes, a calculator, a globe, a microscope, and gears. At the bottom, from left to right, are gears, a DNA double helix, a book, a laptop, and a rocket ship.

LAB REPORT



ULAB
UNIVERSITY OF LIBERAL ARTS
BANGLADESH

OBJECT ORIENTED PROGRAMMING

LAB REPORT 07

Submitted By:

Piyas Sarkar

223014024

Section 04

App class:

```
1 public class App
2 {
3     public static void main( String[] args )
4     {
5         // question 1
6         ArrayList<Vehicle> vehicles = new ArrayList<Vehicle>();
7         Car car1 = new Car("Toyota", "Corolla", 20000, "Red", 4, 5);
8         Car car2 = new Car("Honda", "Civic", 25000, "Blue", 4, 5);
9         Motorcycle motorcycle1 = new Motorcycle("Yamaha", "R1", 15000, "Black", "Sport", 1000);
10        Motorcycle motorcycle2 = new Motorcycle("Harley Davidson", "roadster", 20000, "Orange", "Cruiser", 1200);
11        vehicles.add(car1);
12        vehicles.add(car2);
13        vehicles.add(motorcycle1);
14        vehicles.add(motorcycle2);
15
16        System.out.println("list of Cars");
17        for (Vehicle vehicle : vehicles) {
18            if (vehicle.getClass().getName().equals("oopd7.Car"))
19                System.out.println(vehicle.toString());
20        }
21        System.out.println("-----");
22        System.out.println();
23        System.out.println("list of Motorcycles");
24        for (Vehicle vehicle : vehicles) {
25            if (vehicle.getClass().getName().equals("oopd7.Motorcycle"))
26                System.out.println(vehicle.toString());
27        }
```

```
1 // question 3
2     ArrayList<Employee> employees = new ArrayList<Employee>();
3     Manager manager1 = new Manager("kudus", 101, 50000, 10000);
4     Manager manager2 = new Manager("kislu", 103, 60000, 12000);
5     Engineer engineer1 = new Engineer("mojid", 102, 40000, 5000);
6     Engineer engineer2 = new Engineer("hablu", 104, 45000, 6000);
7
8     employees.add(manager1);
9     employees.add(manager2);
10    employees.add(engineer1);
11    employees.add(engineer2);
12
13    System.out.println("List of Employees");
14    for (Employee employee : employees) {
15        System.out.println(employee.toString());
16        System.out.println("Total Salary: " + employee.calculateTotalSalary());
17    }
18 // question 4
19    ArrayList<Book> books = new ArrayList<Book>();
20    FictionBook fictionBook1 = new FictionBook("The Great Gatsby", "F. Scott Fitzgerald", 10.99, "Classic");
21    NonFictionBook nonFictionBook1 = new NonFictionBook("Sapiens", "Yuval Noah Harari", 15.99, "History");
22
23    books.add(fictionBook1);
24    books.add(nonFictionBook1);
25
26    System.out.println("List of Books");
27    for (Book book : books) {
28        book.displayDetails();
29    }
30
31 }
32
33 }
34
```

```

1 // question 2
2 ArrayList<Shape> shapes = new ArrayList<Shape>();
3 Circle circle1 = new Circle("Circle", "Red", 7);
4 Circle circle2 = new Circle("Circle", "Green", 10);
5 Rectangle rectangle1 = new Rectangle("Rectangle", "Blue", 5, 10);
6 Rectangle rectangle2 = new Rectangle("Rectangle", "Yellow", 8, 12);
7
8 shapes.add(circle1);
9 shapes.add(circle2);
10 shapes.add(rectangle1);
11 shapes.add(rectangle2);
12
13 System.out.println("List of Shapes");
14 for (Shape shape : shapes) {
15     System.out.println(shape.toString());
16     System.out.println("Area: " + shape.calculateArea());
17     System.out.println("Perimeter: " + shape.calculatePerimeter());
18     System.out.println("-----");
19 }

```

Question 01:

Create a class named "vehicle" with properties such as "brand," "model," "price," and "color." Create two subclasses, "car" and "motorcycle," with additional properties specific to each class. Create a main class to initialize objects of the subclasses, store them in an ArrayList, and display their information. Use the 'super' keyword to call the superclass's constructor from the subclasses and utilize method overriding.

Solution code photos:

```
1 package oopd7.vehicle;
2
3 public class Car extends Vehicle {
4     private int doors;
5     private int seats;
6     public Car(String brand, String model, double price, String color, int doors, int seats) {
7         super(brand, model, price, color);
8         this.doors = doors;
9         this.seats = seats;
10    }
11    public int getDoors() {
12        return doors;
13    }
14    public void setDoors(int doors) {
15        this.doors = doors;
16    }
17    public int getSeats() {
18        return seats;
19    }
20    public void setSeats(int seats) {
21        this.seats = seats;
22    }
23    @Override
24    public String toString() {
25        return super.toString() + "\nDoors: " + doors + "\nSeats: " + seats;
26    }
27 }
```

```
1 package oopd7.vehicle;
2
3 public class Car extends Vehicle {
4     private int doors;
5     private int seats;
6     public Car(String brand, String model, double price, String color, int doors, int seats) {
7         super(brand, model, price, color);
8         this.doors = doors;
9         this.seats = seats;
10    }
11    public int getDoors() {
12        return doors;
13    }
14    public void setDoors(int doors) {
15        this.doors = doors;
16    }
17    public int getSeats() {
18        return seats;
19    }
20    public void setSeats(int seats) {
21        this.seats = seats;
22    }
23    @Override
24    public String toString() {
25        return super.toString() + "\nDoors: " + doors + "\nSeats: " + seats;
26    }
27 }
```

```
1 package oopd7.vehicle;
2
3 public class Vehicle {
4     private String brand;
5     private String model;
6     private double price;
7     private String color;
8     public Vehicle(String brand, String model, double price, String color) {
9         this.brand = brand;
10        this.model = model;
11        this.price = price;
12        this.color = color;
13    }
14    public String getBrand() {
15        return brand;
16    }
17
18    public void setBrand(String brand) {
19        this.brand = brand;
20    }
21
22    public String getModel() {
23        return model;
24    }
25
26    public void setModel(String model) {
27        this.model = model;
28    }
29
30    public double getPrice() {
31        return price;
32    }
33
34    public void setPrice(double price) {
35        this.price = price;
36    }
37
38    public String getColor() {
39        return color;
40    }
41
42    public void setColor(String color) {
43        this.color = color;
44    }
45
46    public String toString() {
47        return "Brand: " + brand + "\nModel: " + model + "\nPrice: " + price + "\nColor: " + color;
48    }
49 }
```

Code Outputs:

List of Cars

Brand: Toyota

Model: Corolla

Price: 20000.0

Color: Red

Doors: 4

Seats: 5

Brand: Honda

Model: Civic

Price: 25000.0

Color: Blue

Doors: 4

Seats: 5

List of Motorcycles

Brand: Yamaha

Model: R1

Price: 15000.0

Color: Black

Type: Sport

CC: 1000

Brand: Harley Davidson

Model: roadster

Price: 20000.0

Color: Orange

Type: Cruiser

Question 02:

2. Create a class named "Shape" with properties such as "name" and "color." Create two subclasses, "circle" and "rectangle," with additional properties specific to each class. Implement subclass methods to calculate each shape's area and perimeter. Create objects of the subclasses, store them in an ArrayList, and display their information.

SolutionCodePhotos:



```
1  package oopd7.shape;
2
3  public class Circle extends Shape {
4      private double radius;
5
6      public Circle(String name, String color, double radius) {
7          super(name, color);
8          this.radius = radius;
9      }
10
11     public double getRadius() {
12         return radius;
13     }
14
15     public void setRadius(double radius) {
16         this.radius = radius;
17     }
18
19     @Override
20     public double calculateArea() {
21         return Math.PI * radius * radius;
22     }
23
24     @Override
25     public double calculatePerimeter() {
26         return 2 * Math.PI * radius;
27     }
28
29     @Override
30     public String toString() {
31         return super.toString() + "\nRadius: " + radius;
32     }
33 }
34
```

```
1 package oopd7.shape;
2
3 public class Rectangle extends Shape {
4     private double length;
5     private double width;
6
7     public Rectangle(String name, String color, double length, double width) {
8         super(name, color);
9         this.length = length;
10        this.width = width;
11    }
12
13    public double getLength() {
14        return length;
15    }
16
17    public void setLength(double length) {
18        this.length = length;
19    }
20
21    public double getWidth() {
22        return width;
23    }
24
25    public void setWidth(double width) {
26        this.width = width;
27    }
28
29    @Override
30    public double calculateArea() {
31        return length * width;
32    }
33
34    @Override
35    public double calculatePerimeter() {
36        return 2 * (length + width);
37    }
38
39    @Override
40    public String toString() {
41        return super.toString() + "\nLength: " + length + "\nWidth: " + width;
42    }
43 }
44
```

```
1 package oopd7.shape;
2
3 public abstract class Shape {
4     private String name;
5     private String color;
6
7     public Shape(String name, String color) {
8         this.name = name;
9         this.color = color;
10    }
11
12    public String getName() {
13        return name;
14    }
15
16    public void setName(String name) {
17        this.name = name;
18    }
19
20    public String getColor() {
21        return color;
22    }
23
24    public void setColor(String color) {
25        this.color = color;
26    }
27
28    public abstract double calculateArea();
29    public abstract double calculatePerimeter();
30
31    @Override
32    public String toString() {
33        return "Name: " + name + "\nColor: " + color;
34    }
35 }
36
```

Code Outputs:

List of Shapes

Name: Circle

Color: Red

Radius: 7.0

Area: 153.93804002589985

Perimeter: 43.982297150257104

Name: Circle

Color: Green

Radius: 10.0

Area: 314.1592653589793

Perimeter: 62.83185307179586

Name: Rectangle

Color: Blue

Length: 5.0

Width: 10.0

Area: 50.0

Perimeter: 30.0

Name: Rectangle

Color: Blue

Length: 5.0

Width: 10.0

Area: 50.0

Perimeter: 30.0

Name: Rectangle

Color: Yellow

Length: 8.0

Width: 12.0

Area: 96.0

Perimeter: 40.0

Question 03:

3. Create a class named "Employee" with properties such as "name," "id," and "salary." Create two subclasses, "Manager" and "Engineer," with additional properties specific to each class. Implement a method in each subclass to calculate the total salary by including additional bonuses. Create objects of the subclasses, store them in an ArrayList, and display their information.

Solution code photos:

```
1 package oopd7.Employee;
2
3 public abstract class Employee {
4     private String name;
5     private int id;
6     private double salary;
7
8     public Employee(String name, int id, double salary) {
9         this.name = name;
10        this.id = id;
11        this.salary = salary;
12    }
13
14    public String getName() {
15        return name;
16    }
17
18    public void setName(String name) {
19        this.name = name;
20    }
21
22    public int getId() {
23        return id;
24    }
25
26    public void setId(int id) {
27        this.id = id;
28    }
29
30    public double getSalary() {
31        return salary;
32    }
33
34    public void setSalary(double salary) {
35        this.salary = salary;
36    }
37    public abstract double calculateTotalSalary();
38
39    @Override
40    public String toString() {
41        return "Name: " + name + "\nID: " + id + "\nSalary: " + salary;
42    }
43 }
44
```

```
1 package oopd7.Employee;
2
3 public class Engineer extends Employee {
4     private double bonus;
5
6     public Engineer(String name, int id, double salary, double bonus) {
7         super(name, id, salary);
8         this.bonus = bonus;
9     }
10
11     public double getBonus() {
12         return bonus;
13     }
14
15     public void setBonus(double bonus) {
16         this.bonus = bonus;
17     }
18
19     @Override
20     public double calculateTotalSalary() {
21         return getSalary() + bonus;
22     }
23
24     @Override
25     public String toString() {
26         return super.toString() + "\nBonus: " + bonus;
27     }
28 }
29
```



```
1  package oopd7.Employee;
2
3  public class Manager extends Employee {
4      private double bonus;
5
6      public Manager(String name, int id, double salary, double bonus) {
7          super(name, id, salary);
8          this.bonus = bonus;
9      }
10
11     public double getBonus() {
12         return bonus;
13     }
14
15     public void setBonus(double bonus) {
16         this.bonus = bonus;
17     }
18
19     @Override
20     public double calculateTotalSalary() {
21         return getSalary() + bonus;
22     }
23
24     @Override
25     public String toString() {
26         return super.toString() + "\nBonus: " + bonus;
27     }
28 }
29
```

Code Outputs:

Name: Rectangle

Color: Blue

Length: 5.0

Width: 10.0

Area: 50.0

Perimeter: 30.0

Name: Rectangle

Color: Yellow

Length: 8.0

Width: 12.0

Area: 96.0

Perimeter: 40.0

List of Employees

Name: kodus

ID: 101

Salary: 50000.0

Bonus: 10000.0

Total Salary: 60000.0

Name: kislui

ID: 103

Salary: 60000.0

Bonus: 12000.0

Total Salary: 72000.0

Name: mojid

ID: 102

Salary: 40000.0

Bonus: 5000.0

Total Salary: 45000.0

Name: hablu

ID: 104

Salary: 45000.0


Bonus: 6000.0

Total Salary: 51000.0

Question 04:

4. Create a class named "Book" with properties such as "title," "author," and "price." Create two subclasses, "FictionBook" and "NonFictionBook," with additional properties specific to each class. Implement methods in the subclass to display the book details and perform book-specific actions. Create objects of the subclasses, store them in an ArrayList, and display their information.

Solution code photos:



```
1  package oopd7.BOOK;
2
3  public abstract class Book {
4      private String title;
5      private String author;
6      private double price;
7
8      public Book(String title, String author, double price) {
9          this.title = title;
10         this.author = author;
11         this.price = price;
12     }
13
14     public String getTitle() {
15         return title;
16     }
17
18     public void setTitle(String title) {
19         this.title = title;
20     }
21
22     public String getAuthor() {
23         return author;
24     }
25
26     public void setAuthor(String author) {
27         this.author = author;
28     }
29
30     public double getPrice() {
31         return price;
32     }
33
34     public void setPrice(double price) {
35         this.price = price;
36     }
37
38     // Abstract method to display book details
39     public abstract void displayDetails();
40 }
41
```



```
1 package oopd7.BOOK;
2
3 public class FictionBook extends Book {
4     private String genre;
5
6     public FictionBook(String title, String author, double price, String genre) {
7         super(title, author, price);
8         this.genre = genre;
9     }
10
11     public String getGenre() {
12         return genre;
13     }
14
15     public void setGenre(String genre) {
16         this.genre = genre;
17     }
18
19     @Override
20     public void displayDetails() {
21         System.out.println("Title: " + getTitle());
22         System.out.println("Author: " + getAuthor());
23         System.out.println("Price: $" + getPrice());
24         System.out.println("Genre: " + genre);
25     }
26 }
27
```



```
1 package oopd7.BOOK;
2
3 public class NonFictionBook extends Book {
4     private String subject;
5
6     public NonFictionBook(String title, String author, double price, String subject) {
7         super(title, author, price);
8         this.subject = subject;
9     }
10
11     public String getSubject() {
12         return subject;
13     }
14
15     public void setSubject(String subject) {
16         this.subject = subject;
17     }
18
19     @Override
20     public void displayDetails() {
21         System.out.println("Title: " + getTitle());
22         System.out.println("Author: " + getAuthor());
23         System.out.println("Price: $" + getPrice());
24         System.out.println("Subject: " + subject);
25     }
26 }
27
```

Code Outputs:

List of Books

Title: The Great Gatsby

Author: F. Scott Fitzgerald

Price: \$10.99

Genre: Classic

Title: Sapiens

Author: Yuval Noah Harari

Price: \$15.99

Subject: History