

1)

```
class Shape {

    private int dimension;

    Shape(int dimension) {
        this.dimension = dimension;
    }

}

class Circle extends Shape {

    private int radius;

    Circle(int radius) {
        super(radius);
        System.out.println(3.14 * radius * radius);
    }

}

class Rectangle extends Shape {

    private int length;
    private int breadth;

    Rectangle(int length, int breadth) {
        super(length);
        System.out.println(length * breadth);
    }

}

public class Assignment {

    public static void main(String[] args) {
        Circle c1 = new Circle(10);
        Rectangle r1 = new Rectangle(10, 20);
    }

}
```

2)

```
class Student {

    String name;
    String gender;
```

```

    public Student(String name, String gender) {
        this.name = name;
        this.gender = gender;
    }

    @Override
    public String toString() {
        return "Student [name=" + name + ", gender=" + gender + "]";
    }
}

class ZSGSStudent extends Student {

    private int zsgsID;
    public ZSGSStudent(int zsgsID, String name, String gender) {
        super(name, gender);
        this.zsgsID = zsgsID;
    }

    @Override
    public String toString() {
        return "ZSGSStudent [zsgsID=" + zsgsID + "]";
    }
}

}

public class Assignment    {
    public static void main(String[] args) {
        Student s = new Student("Dinesh", "Male");
        ZSGSStudent z = new ZSGSStudent(2, "Kumari", "Female");

        System.out.println(s.toString());
        System.out.println(z.toString());
    }
}

```

3)

```

class Vehicle {

    String model;

```

```

    int manufactureYear;
    int odometerReading;

    public Vehicle(String model, int manufactureYear, int odometerReading) {
        this.model = model;
        this.manufactureYear = manufactureYear;
        this.odometerReading = odometerReading;
    }

    @Override
    public String toString() {
        return "The vehicle is " + model + " and manufactured in " + manufactureYear + "
it has ran " + odometerReading;
    }
}

class Car extends Vehicle    {

    int peopleCapacity;

    public Car(String model, int manufactureYear, int odometerReading, int peopleCapacity)
{
        super(model, manufactureYear, odometerReading);
        this.peopleCapacity = peopleCapacity;
    }

    public String toString()    {
        return super.toString() + ". The car can carry " + peopleCapacity + " people.";
    }
}

class Bike extends Vehicle    {

    int peopleCapacity;

    public Bike(String model, int manufactureYear, int odometerReading, int peopleCapacity)
{
        super(model, manufactureYear, odometerReading);
        this.peopleCapacity = peopleCapacity;
    }

    public String toString()    {

```

```

        return super.toString() + ". The bike can carry" + peopleCapacity + " people";
    }

}

public class Assignment    {
    public static void main(String[] args) {
        Car c = new Car("Sedan", 2017, 2490, 5);
        Bike b = new Bike("Commuter", 2022, 1031, 2);

        System.out.println(c.toString());
        System.out.println(b.toString());
    }
}

```

4)

```
package ZSGS;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
class LibraryManagementSystem {
    private String userType;
    private String username;
    private String password;

    public void login() {
        System.out.println("Logged in successfully.");
    }

    public void register() {
        System.out.println("Registered successfully.");
    }

    public void logout() {
        System.out.println("Logged out successfully.");
    }
}

```

```
class User {
    private String name;
    private int id;
}

```

```

public User(String name, int id) {
    this.name = name;
    this.id = id;
}

public boolean verify() {
    System.out.println("User verified.");
    return true;
}

public void checkAccount() {
    System.out.println("Account checked.");
}

public void getBookInfo() {
    System.out.println("Book information retrieved.");
}

@Override
public String toString() {
    return "User [name=" + name + ", id=" + id + "]";
}
}

class Librarian extends User {
    private String password;
    private String searchString;

    public Librarian(String name, int id, String password) {
        super(name, id);
        this.password = password;
    }

    public boolean verifyLibrarian() {
        System.out.println("Librarian verified.");
        return true;
    }

    public void search(String searchString) {
        System.out.println("Searching for: " + searchString);
    }
}

class Book {

```

```

private String title;
private String author;
private String ISBN;
private String publication;

public Book(String title, String author, String ISBN, String publication) {
    this.title = title;
    this.author = author;
    this.ISBN = ISBN;
    this.publication = publication;
}

public void showDueDate() {
    System.out.println("Due date shown.");
}

public void reservationStatus() {
    System.out.println("Reservation status shown.");
}

public void feedback() {
    System.out.println("Feedback provided.");
}

public void bookRequest() {
    System.out.println("Book requested.");
}

public void renewInfo() {
    System.out.println("Renewal information shown.");
}

@Override
public String toString() {
    return "Book [title=" + title + ", author=" + author + ", ISBN=" + ISBN + ", publication=" +
publication + "]";
}
}

class Account {
    private int noBorrowedBooks;
    private int noReservedBooks;
    private int noReturnedBooks;
    private int noLostBooks;

```

```

private double fineAmount;

public Account() {
    this.noBorrowedBooks = 0;
    this.noReservedBooks = 0;
    this.noReturnedBooks = 0;
    this.noLostBooks = 0;
    this.fineAmount = 0.0;
}

public void calculateFine() {
    System.out.println("Fine calculated: $" + fineAmount);
}

}

class LibraryDatabase {
    private List<Book> listOfBooks;

    public LibraryDatabase() {
        listOfBooks = new ArrayList<>();
    }

    public void add(Book book) {
        listOfBooks.add(book);
        System.out.println("Book added: " + book);
    }

    public void delete(Book book) {
        listOfBooks.remove(book);
        System.out.println("Book deleted: " + book);
    }

    public void update(int index, Book book) {
        listOfBooks.set(index, book);
        System.out.println("Book updated at index " + index + ": " + book);
    }

    public void display() {
        System.out.println("Library Books:");
        for (Book book : listOfBooks) {
            System.out.println(book);
        }
    }
}

```

```

        public void search(String searchString) {
            System.out.println("Searching for books with: " + searchString);
        }
    }

    class Staff extends User {
        private String dept;

        public Staff(String name, int id, String dept) {
            super(name, id);
            this.dept = dept;
        }

        @Override
        public String toString() {
            return "Staff [dept=" + dept + ", " + super.toString() + "]";
        }
    }

    class Student extends User {
        private String studentClass;

        public Student(String name, int id, String studentClass) {
            super(name, id);
            this.studentClass = studentClass;
        }

        @Override
        public String toString() {
            return "Student [class=" + studentClass + ", " + super.toString() + "]";
        }
    }

    public class Assignment {
        public static void main(String[] args) {
            LibraryManagementSystem lms = new LibraryManagementSystem();
            lms.login();
            lms.register();
            lms.logout();

            User user = new User("Dinesh", 1);
            user.verify();
            user.checkAccount();
            user.getBookInfo();
        }
    }

```



```

Librarian librarian = new Librarian("Bob", 2, "password123");
librarian.verifyLibrarian();
librarian.search("Harry Potter");

Book book1 = new Book("Harry Potter", "J.K. Rowling", "1234567890", "Bloomsbury");
Book book2 = new Book("Percy Jackson", "Rick Riordan", "0987654321",
"Disney-Hyperion");

book1.showDueDate();
book1.reservationStatus();
book1.feedback();
book1.bookRequest();
book1.renewInfo();

Account account = new Account();
account.calculateFine();

LibraryDatabase libraryDatabase = new LibraryDatabase();
libraryDatabase.add(book1);
libraryDatabase.add(book2);
libraryDatabase.display();
libraryDatabase.search("Percy Jackson");

Staff staff = new Staff("Kumar", 3, "Library");
Student student = new Student("Mooley", 4, "10th Grade");

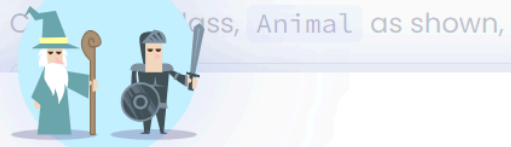
System.out.println(staff);
System.out.println(student);
}
}

```

5)

solutions

ve solved this exercise and  
[Community Solutions.](#)



# You've completed Wizards and Warriors!

Awesome work. You're one step closer to learning Java 🚀

You've learnt 1 concept by completing this exercise.



Inheritance

