ONLINE WARDROBE APP

ELITE THREADS

JAVA PROJECT

ANGELINE A

2348409

TABLE OF CONTENTS

01

INTRODUCTION

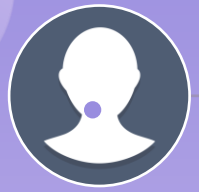
PACKAGES, FILES AND
THEIR CONNECTION

PACKAGES (FILES)

1. User
2. Item
3. Product
4. Category
5. Admin
6. AccountManagement
7. Customer
8. KidsCategory
9. KidsItemType
10. MenItemType
11. WomenItemType
12. WomenWardrobe
13. MenWardrobe
14. KidsWardrobe
15. WomenWardrobe
16. ShoppingCart
17. ShoppingApp

CONNECTIONS

USER



Admin
Customer
Main

PRODUCT

ShoppingApp
ShoppingCart

CUSTOMER

ShoppingApp
Main

CATEGORY

ShoppingApp

CONNECTIONS

ACCOUNTMANAGEMENT

Admin
ShoppingApp

KIDSCATEGORY

ShoppingApp
KidsWardrobe

KIDSITEM

KidsWardrobe

KIDSWARDROBE

ShoppingApp

CONNECTIONS

MENITEMTYPE

MenWardrobe

MENWARDROBE

ShoppingApp

WOMENITEMTYPE

WomenWardrobe

WOMENWARDROBE

ShoppingApp

CONNECTIONS

SHOPPINGAPP

Main

SHOPPINGCART

ShoppingApp
ShoppingCart
MenWardrobe
KidsWardrobe
WomenWardrobe

ITEM

ShoppingApp
ShoppingCart
MenWardrobe
KidsWardrobe
WomenWardrobe

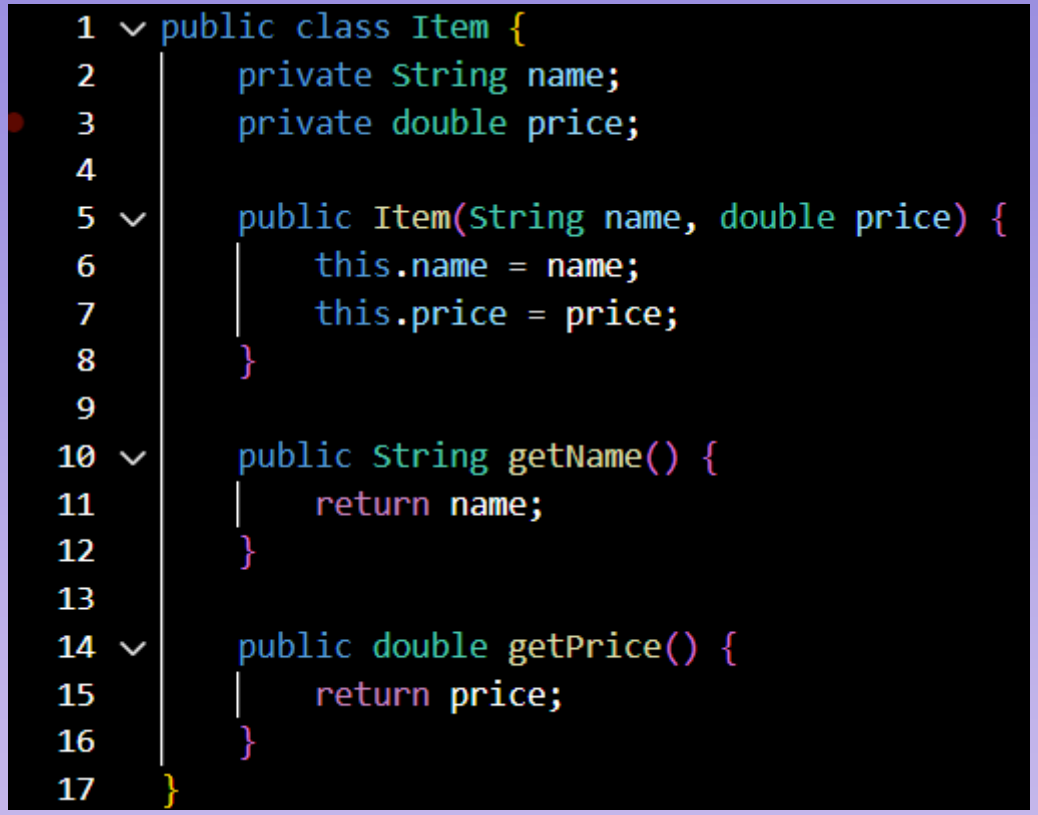
02

IMPLEMENTATION

IMPLEMENTATION OF JAVA CONCEPTS

ENCAPSULATION

- **Explanation:** Encapsulation is the concept of bundling data (attributes) and methods (behavior) that operate on the data into a single unit called a class. It hides the internal state of an object from the outside world and only exposes the necessary functionality through methods.
- **Working in the Application:** In the shopping application, various classes such as Item, Category, User, etc., encapsulate their attributes (like name, price, username, etc.) and methods (like adding items to the cart, calculating total amount, etc.). This ensures data integrity and reduces the risk of unintended access or modification.


public class Item {
private String name;
private double price;
public Item(String name, double price) {
this.name = name;
this.price = price;
}
public String getName() {
return name;
}
public double getPrice() {
return price;
}
}

INHERITANCE

- **Explanation:** Inheritance is a mechanism in which a new class (subclass) is derived from an existing class (superclass). The subclass inherits the attributes and methods of the superclass and can also have its own unique attributes and methods.
- **Working in the Application:** The application uses inheritance to create different types of wardrobes for men, women, and kids. For example, MenWardrobe and WomenWardrobe inherit from a common superclass Wardrobe, which contains common methods for browsing items. This promotes code reuse and helps in maintaining a consistent structure across different wardrobe types.

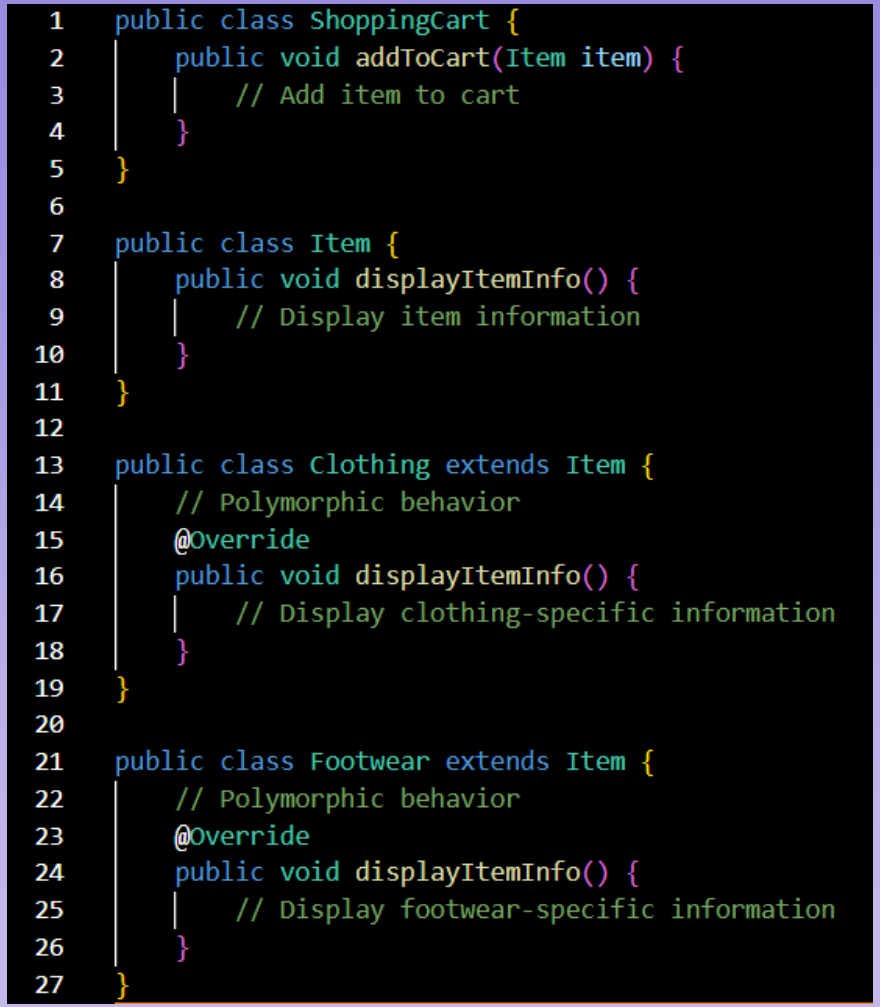
```java
public class Wardrobe {
    public void browseWardrobe() {
        System.out.println("Browsing wardrobe...");
    }
}

public class MenWardrobe extends Wardrobe {
    // Additional methods specific to men's wardrobe
}

public class WomenWardrobe extends Wardrobe {
    // Additional methods specific to women's wardrobe
}
```

POLYMORPHISM

- **Explanation:** Polymorphism allows objects of different classes to be treated as objects of a common superclass. It enables methods to behave differently based on the object they are called on, leading to flexibility and extensibility in the code.
- **Working in the Application:** Polymorphism is demonstrated when different categories of items (men's clothing, women's clothing, etc.) are added to the shopping cart using a common method addToCart() from the ShoppingCart class. The specific behavior of adding an item is determined at runtime based on the type of item being added.



```java
public class ShoppingCart {
    public void addToCart(Item item) {
        // Add item to cart
    }
}

public class Item {
    public void displayItemInfo() {
        // Display item information
    }
}

public class Clothing extends Item {
    // Polymorphic behavior
    @Override
    public void displayItemInfo() {
        // Display clothing-specific information
    }
}

public class Footwear extends Item {
    // Polymorphic behavior
    @Override
    public void displayItemInfo() {
        // Display footwear-specific information
    }
}
```

ABSTRACTION

- **Explanation:** Abstraction involves hiding complex implementation details and exposing only the essential features of an object. It allows programmers to focus on what an object does rather than how it does it.
- **Working in the Application:** The application abstracts away the internal implementation of shopping cart operations, such as adding items, calculating total amounts, and generating invoices. Users interact with high-level methods and interfaces without needing to understand the underlying complexities.

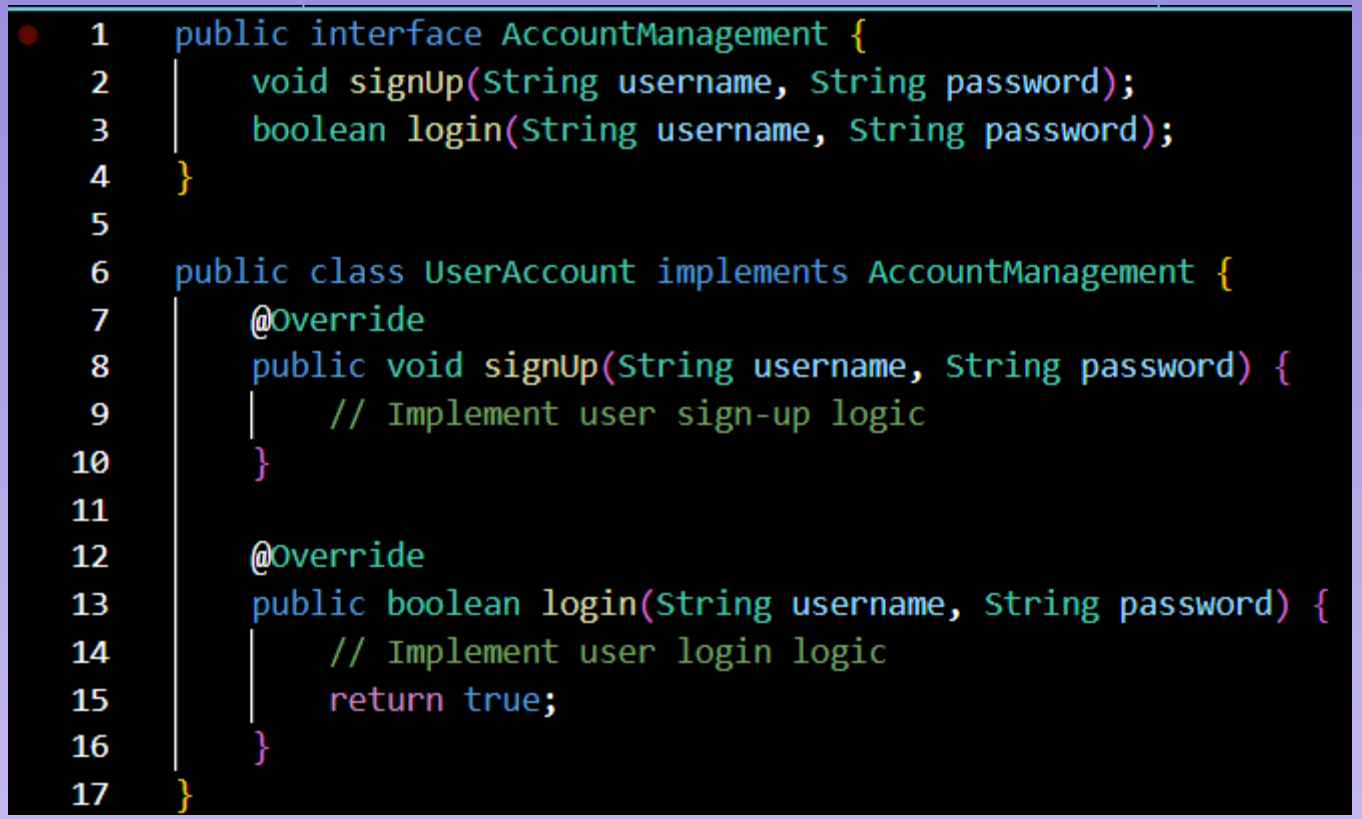
```java
public interface PaymentGateway {
    void processPayment(double amount);
}

public class OnlinePaymentGateway implements PaymentGateway {
    @Override
    public void processPayment(double amount) {
        // Implement online payment processing logic
    }
}

public class CashOnDeliveryGateway implements PaymentGateway {
    @Override
    public void processPayment(double amount) {
        // Implement cash on delivery payment processing logic
    }
}
```

INTERFACE

- **Explanation:** Interfaces define a contract for classes to implement, specifying a set of methods that must be implemented by any class that implements the interface. This promotes loose coupling and allows classes to work with objects of different types through a common interface.
- **Working in the Application:** Interfaces like AccountManagement define methods for user authentication (signUp() and login()). Various classes such as ShoppingApp implement this interface, providing their own implementations of these methods. This allows for flexibility in authentication mechanisms and promotes code modularization.


1 public interface AccountManagement {
2 void signUp(String username, String password);
3 boolean login(String username, String password);
4 }
5
6 public class UserAccount implements AccountManagement {
7 @Override
8 public void signUp(String username, String password) {
9 // Implement user sign-up logic
10 }
11
12 @Override
13 public boolean login(String username, String password) {
14 // Implement user login logic
15 return true;
16 }
17 }

MODULARITY

- **Explanation:** Modularity is the division of a software system into separate, independently replaceable modules. It promotes code organization, reusability, and easier maintenance by breaking down complex systems into smaller, manageable components.
- **Working in the Application:** The application is divided into modular components such as wardrobes for different categories, shopping cart functionalities, user authentication, and payment processing. Each module encapsulates related functionality, making it easier to understand, modify, and extend the system.

GENERICS

```java
import java.util.*;

public class ShoppingCart<T extends Item> {
    private List<T> items;

    public ShoppingCart() {
        this.items = new ArrayList<>();
    }

    public void addItem(T item) {
        items.add(item);
    }

    public List<T> getItems() {
        return items;
    }
}
```

MULTITHREADING

```java
public class PaymentThread extends Thread {
    private ShoppingCart cart;

    public PaymentThread(ShoppingCart cart) {
        this.cart = cart;
    }

    @Override
    public void run() {
        System.out.println("Processing payment...");
        try {
            // Simulate payment processing time
            Thread.sleep(3000); // Sleep for 3 seconds
            System.out.println("Payment completed successfully.");
            // Update payment status in the shopping cart
            cart.setPaymentStatus(true);
        } catch (InterruptedException e) {
            System.out.println("Payment process interrupted.");
        }
    }
}
```

COLLECTION

```java
public class MenWardrobe {
    private Map<Category, List<Item>> wardrobe;

    public MenWardrobe() {
        this.wardrobe = new HashMap<>();
    }

    public void initializeWardrobe() {
        // Initialize wardrobe with categories and items
        List<Item> shirts = new ArrayList<>();
        shirts.add(new Item("Striped Shirt", 25.0));
        shirts.add(new Item("Polo T-shirt", 20.0));

        wardrobe.put(Category.SHIRTS, shirts);

        // Add more categories and items as needed
    }
}
```

EXCEPTIONAL HANDLING

```java
public class AccountManagement {
    public boolean login(String username, String password) throws InvalidCredentialsException {
        if (username == null || password == null) {
            throw new NullPointerException("Username or password cannot be null.");
        }
        if (!userCredentials.containsKey(username) || !userCredentials.get(username).equals(password)) {
            throw new InvalidCredentialsException("Invalid username or password.");
        }
        return true;
    }
}
```

03

WORKING

APPLICATION FEATURES,
WORKING OF THE CODE
AND OUTPUT

APPLICATION FEATURES

- User Authentication and Signup
- Browsing Product Categories
- Adding Items to the Cart
- Finalizing Purchase
- Optional Premium Membership Upgrade
- Invoice and delivery Tracking

OUTPUT

```
D:\JAVA Programs\Elite Thread Online Wardrobe>java Main.java
================================================
=== Welcome to ELITE THREADS Online Shopping App! ===
 __        __   _
 \ \      / /__| | ___ ___  _ __ ___   ___
  \ \ /\ / / _ \ |/ __/ _ \| '_ ` _ \ / _ \
   \ V  V /  __/ | (_| (_) | | | | | |  __/
    \_/\_/ \___|_|\___\___/|_| |_| |_|\___|

==== Please Sign up if you are new user ====
1. Sign Up
2. Login
3. Exit
Enter your choice: 1
==== Sign Up ====
Enter username: Angeline
Enter gender: Female
Enter age: 21
Enter password: elite
Confirm password: elite
Account successfully created.

==== Login ====
Enter username: Angeline
Enter password: elite
======== ELITE THREADS ========
1. Browse Wardrobe
2. View Cart
3. Finalize Products
4. Update Premium Membership
5. Logout
Enter your choice: 1
==== ELITE WARDROBE =====
==== PARTNERS FROM LEADING BRANDS ====
==== CLOTHING ====
Zara, H&M, Forever 21, Gap, Nike, Adidas, Levi's, Ralph Lauren, Tommy Hilfiger, Calvin Klein

==== JEWLLERY ====
Tiffany & Co., Pandora, Swarovski, Cartier, David Yurman, Alex and Ani, Kendra Scott, Bulgari, Chopard, Van Cleef & Arpels

==== ACCESSOREIES ====
Coach, Michael Kors, Kate Spade, Louis Vuitton, Gucci, Fossil, Titan, , Ray-Ban, Oakley, Prada, Hermès

==== FOOTWEAR ====
Nike, Adidas, Puma, Reebok, Converse, Vans, Timberland, Dr. Martens, Birkenstock, Skechers
```

OUTPUT

```
======= EXCITING COLLECTIONS AHEAD ======

1. MEN
2. WOMEN
3. KIDS
4. Go back to previous options

Enter your choice: 2
==== Elevate your elegance, Embrace your unique allure ====

==== WOMEN ====

1. FOOTWEAR
2. ACCESSORIES
3. ETHNIC
4. JEWELLERY
5. WESTERN
6. FORMALS
7. View Cart
8. Go back to previous options

3
===== Available items =====
1. Traditional Saree - Kanjivaram Silk - Red - $150.0
2. Anarkali Suit - Designer - Blue - $200.0
3. Lehenga - Embroidered - Green - $180.0
4. Salwar Kameez - Cotton - Pink - $160.0
5. Kurti - Floral Print - Yellow - $100.0
6. Churidar - Silk - Orange - $170.0
7. Saree - Georgette - Purple - $120.0
8. Ethnic Gown - Sequin - Silver - $250.0
9. Patiala Suit - Punjabi - Maroon - $190.0
10. Bridal Lehenga - Velvet - Gold - $350.0
11. Go back to Women Wardrobe

4
Salwar Kameez - Cotton - Pink added to cart.
```

```
==== WOMEN ====

1. FOOTWEAR
2. ACCESSORIES
3. ETHNIC
4. JEWELLERY
5. WESTERN
6. FORMALS
7. View Cart
8. Go back to previous options

4
===== Available items =====
1. Ring - Diamond - Solitaire - Gold - $300.0
2. Pendant - Gemstone - Ruby - Silver - $250.0
3. Bracelet - Pearl - Beaded - White - $180.0
4. Earrings - Hoops - Diamond - Rose Gold - $220.0
5. Necklace - Emerald - Choker - Gold - $400.0
6. Anklet - Silver - Traditional - Red - $120.0
7. Bangles - Glass - Bangles - Green - $80.0
8. Brooch - Floral - Brooch - Blue - $150.0
9. Toe Ring - Silver - Toe Ring - Pink - $50.0
10. Hairpin - Rhinestone - Hairpin - Purple - $40.0
11. Go back to Women Wardrobe

2
Pendant - Gemstone - Ruby - Silver added to cart.
```

OUTPUT

```
==== WOMEN ====

1. FOOTWEAR
2. ACCESSORIES
3. ETHNIC
4. JEWELLERY
5. WESTERN
6. FORMALS
7. View Cart
8. Go back to previous options

7
==== Items in Cart: ====
Salwar Kameez - Cotton - Pink - $160.0

Pendant - Gemstone - Ruby - Silver - $250.0

==== WOMEN ====

1. FOOTWEAR
2. ACCESSORIES
3. ETHNIC
4. JEWELLERY
5. WESTERN
6. FORMALS
7. View Cart
8. Go back to previous options

8
======== ELITE THREADS ========
1. Browse Wardrobe
2. View Cart
3. Finalize Products
4. Update Premium Membership
5. Logout
Enter your choice:
```

```
======== ELITE THREADS ========
1. Browse Wardrobe
2. View Cart
3. Finalize Products
4. Update Premium Membership
5. Logout
Enter your choice: 2
==== Items in Cart: ====
Salwar Kameez - Cotton - Pink - $160.0

Pendant - Gemstone - Ruby - Silver - $250.0
```

```
======== ELITE THREADS ========
1. Browse Wardrobe
2. View Cart
3. Finalize Products
4. Update Premium Membership
5. Logout
Enter your choice: 3
==== Items in Cart: ====
==== Items in Cart: ====
Salwar Kameez - Cotton - Pink - $160.0

Pendant - Gemstone - Ruby - Silver - $250.0
```

OUTPUT

```
Do you want to continue with the placement of order? (yes/no)
yes
==== Your total bill: ====
Total Amount: $410.0
GST: $73.8
Delivery Charges: $10.0
Final Amount (Including GST and Delivery Charges): $493.8
Do you want to proceed with the payment? (yes/no)
yes
=== Please provide your address details: ===
Door No: 12
Street: 5th Street
City: Bangalore
State: Karnataka
Country: India
PIN: 56
=== Choose payment option: ===
1. Online
2. Cash on Delivery

Enter your choice: 1
=== Enter card details ===
Card Holder Name: Angeline
Card Number (12 digits): 123456789012
CVV: 124
=== Payment completed successfully ===
=== Your order placed successfully ===
=== Your order will be delivered on: 4 MAY (Saturday) ===
```

```
====================================
==== Invoice ====
Products Purchased:
Salwar Kameez - Cotton - Pink - $160.0
Pendant - Gemstone - Ruby - Silver - $250.0
Total Amount: $410.0
GST: $73.8
Delivery Charges: $10.0
Final Amount (Including GST and Delivery Charges): $493.8
Payment Method: Online
Address:
Door No: 12
Street: 5th Street
City: Bangalore
State: Karnataka
Country: India
PIN: 56
Delivery Date: 4 MAY (Saturday)
====================================
```

OUTPUT

```
=== Rate us! Your stars guide our way! ===
Rate our app (1-5):
4
Thank you for your rating! We'll continue to enhance our app.
```

```
======== ELITE THREADS ========
1. Browse Wardrobe
2. View Cart
3. Finalize Products
4. Update Premium Membership
5. Logout
Enter your choice: 5
Logging out...
==== Please Sign up if you are new user ====
1. Sign Up
2. Login
3. Exit
Enter your choice: 3
=== THANK YOU, VISIT AGAIN ===
** FOLLOW US ON:**
** INSTAGRAM: @elitethreads
** FACEBOOK: @elitethreadsbrd
** OFFICIAL WEBSITE: **
** @elitethreads.com
********Exiting********
```

04

REFERENCES

REFERENCES

[1] Book- https://github.com/EbookFoundation/free-programming-books/blob/main/books/free-programming-books-langs.md

[2] GitHub link - https://devfreebooks.github.io/java/

[3] Generics - https://www.tutorialspoint.com/java/java_generics.htm

[4] Great Learning - https://www.mygreatlearning.com/blog/oops-concepts-in-java/

[5] Oracle Java Documentation - https://docs.oracle.com/en/java/

[6] Java Tutorials - https://docs.oracle.com/javase/tutorial/

THANK YOU!