SCD-Lab

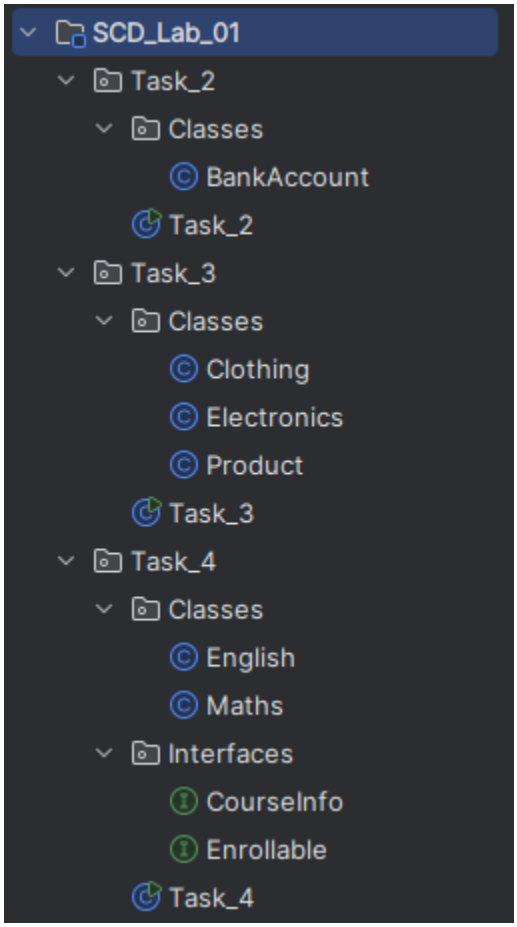
Lab#01 - Home Task

Name: Anas-Altaf

Roll.No: 22F-3639

Java Codes:

File Structure:

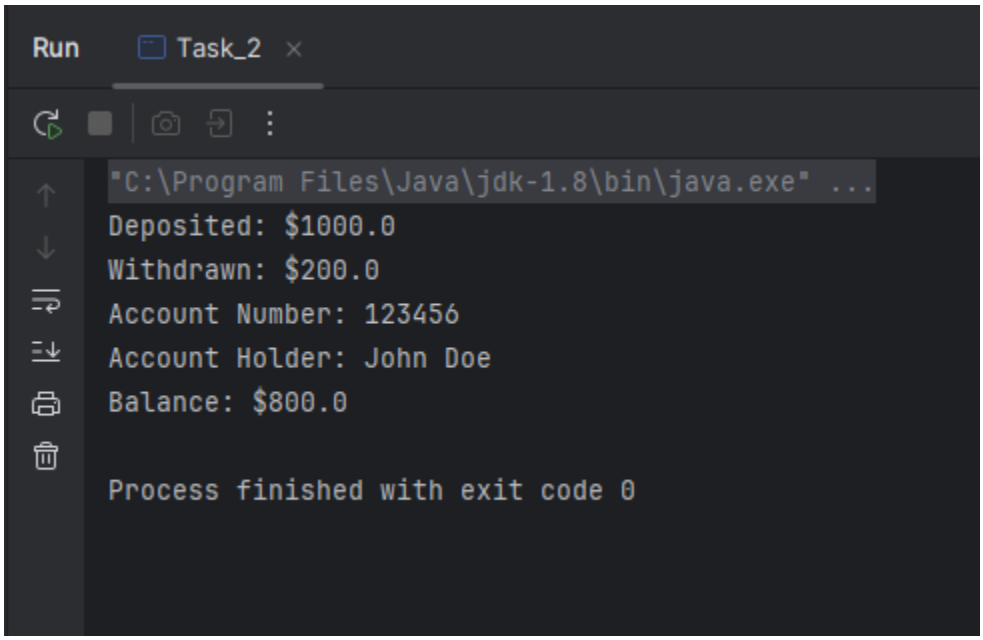


Task-2

```java
package Task_2;

public class BankAccount {
   private int accountNumber;
   private String accountHolderName;
   private double balance;

   // Constructor
   public BankAccount() {
       accountNumber = 0;
       balance = 0.0;
   }

   //Getter Methods
   public int getAccountNumber() {
       return accountNumber;
   }

   public void setAccountNumber(int accountNumber) {
       this.accountNumber = accountNumber;
   }
   //Setter Methods

   public String getAccountHolderName() {
       return accountHolderName;
   }

   public void setAccountHolderName(String accountHolderName) {
       this.accountHolderName = accountHolderName;
   }

   public double getBalance() {
       return balance;
   }

   public void deposit(double amount) {
       if (amount > 0) {
           balance += amount;
           System.out.println("Deposited: $" + amount);
       } else {
           System.out.println("Invalid deposit amount!");
       }
   }

   public void withdraw(double amount) {
       if (amount > 0 && amount <= balance) {
           balance -= amount;
```

```
            System.out.println("Withdrawn: $" + amount);
        } else {
            System.out.println("Invalid or insufficient balance for
withdrawal!");
        }
    }

    public void displayAccountDetails() {
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Account Holder: " + accountHolderName);
        System.out.println("Balance: $" + balance);
    }
}
public class Task_2 {
    public static void main(String[] args) {

        BankAccount account = new BankAccount();

        // Setting account details
        account.setAccountNumber(123456);
        account.setAccountHolderName("John Doe");

        // Depositing and withdrawing funds
        account.deposit(1000.00);
        account.withdraw(200.00);

        // Displaying account details
        account.displayAccountDetails();
    }
}
```

Output:



Task-3

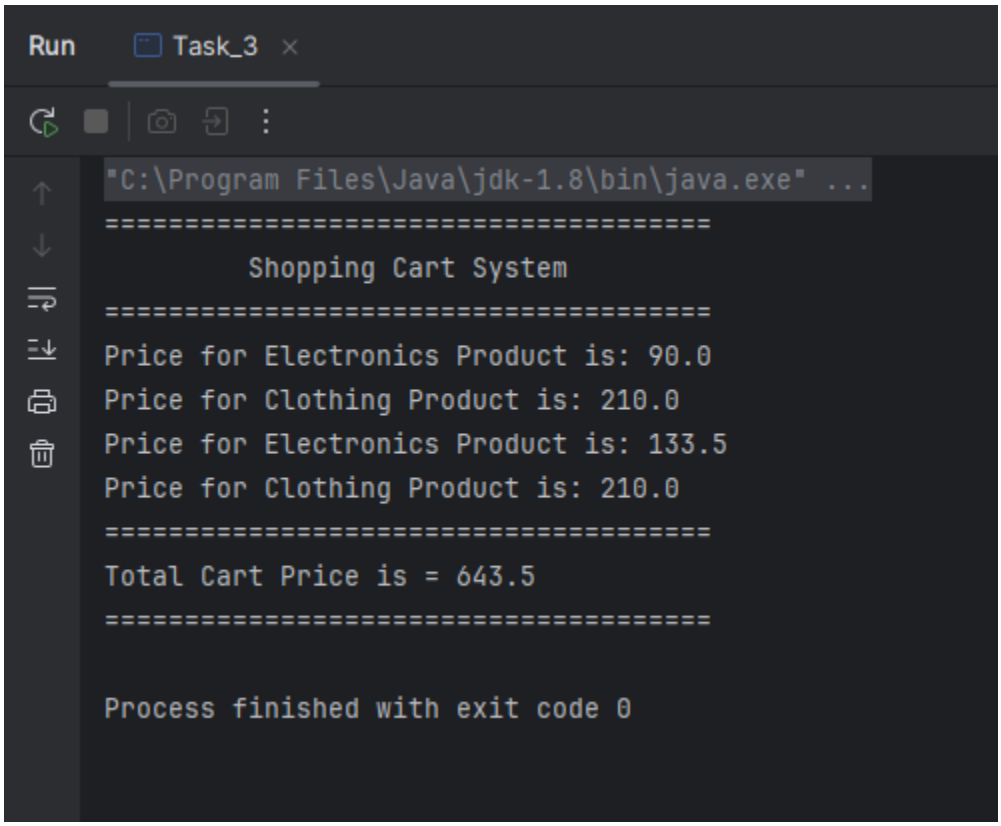
```
package Task_3.Classes;

public class Clothing extends Product {
    private double price;
    private double discountPercentage;

    public Clothing(double price, double discountPercentage) {
        this.price = price;
        this.discountPercentage = discountPercentage;
    }

    @Override
    public double calculatePrice() {
        double discountedPrice = price - (price * (discountPercentage / 100));
        System.out.println("Price for Clothing Product is: " + discountedPrice);
        return discountedPrice;
    }
}
package Task_3.Classes;

public class Electronics extends Product {
    private double price;
    private double discountPercentage;

    public Electronics(double price, double discountPercentage) {
        this.price = price;
```

```
        this.discountPercentage = discountPercentage;
    }

    @Override
    public double calculatePrice() {
        double discountedPrice = price - (price * (discountPercentage / 100));
        System.out.println("Price for Electronics Product is: " +
discountedPrice);
        return discountedPrice;
    }
}
```

```
package Task_3.Classes;

public class Product {
    public double calculatePrice() {
        return 0.0;
    }

}
```

```
package Task_3;

import Task_3.Classes.Clothing;
import Task_3.Classes.Electronics;
import Task_3.Classes.Product;

public class Task_3 {
    public static void main(String[] args) {
        double totalPrice = 0.0;
        Product[] products = new Product[]{
                new Electronics(100, 10),
                new Clothing(300, 30),
                new Electronics(150, 11),
                new Clothing(300, 30),
        };
        System.out.println("=====================================");
        System.out.println("         Shopping Cart System         ");
        System.out.println("=====================================");
        for (int i = 0; i < products.length; i++) {
            totalPrice += products[i].calculatePrice();

        }
        System.out.println("=====================================");
        System.out.println("Total Cart Price is = " + totalPrice);
        System.out.println("=====================================");
    }
}
```

Output:



Task-4

Classes:

```java
package Task_4.Classes;

import Task_4.Interfaces.CourseInfo;
import Task_4.Interfaces.Enrollable;

public class English implements Enrollable, CourseInfo {
    private String courseCode;
    private String courseName;
    private int availableSeats;
    private int totalSeats;

    public English(String courseCode, String courseName, int totalSeats) {
        this.courseCode = courseCode;
        this.courseName = courseName;
        this.availableSeats = this.totalSeats = totalSeats;
    }

    @Override
```

```
    public void getCourseName() {
        System.out.println("Course Name: " + courseName);
    }

    @Override
    public void getCourseCode() {
        System.out.println("Course Code: " + courseCode);
    }

    @Override
    public void getAvailableSeats() {
        System.out.println("Course Available Seats: " + availableSeats);

    }

    @Override
    public void enrollCourse() {
        if (availableSeats > 0) {

            availableSeats--;
            System.out.println("✅| " + courseCode + " | " + courseName + " |
Seats : " + availableSeats + "/" + totalSeats + " | Enrolled!");
        } else {
            System.out.println("⚠| " + courseCode + " | " + courseName + " |
Seats : " + availableSeats + "/" + totalSeats + " | Full!");

        }
    }

    public void dropCourse() {
        if (availableSeats != totalSeats) {

            availableSeats++;
            System.out.println("❌| " + courseCode + " | " + courseName + " |
Seats : " + availableSeats + "/" + totalSeats + " | Dropped!");
        } else {
            System.out.println("⚠| " + courseCode + " | " + courseName + " |
Seats : " + availableSeats + "/" + totalSeats + " | No Enrollments!");

        }
    }
}
```

```
package Task_4.Classes;


import Task_4.Interfaces.CourseInfo;
import Task_4.Interfaces.Enrollable;
```

```java
public class Maths implements Enrollable, CourseInfo {
   private String courseCode;
   private String courseName;
   private int availableSeats;
   private int totalSeats;

   public Maths(String courseCode, String courseName, int totalSeats) {
       this.courseCode = courseCode;
       this.courseName = courseName;
       this.availableSeats = this.totalSeats = totalSeats;
   }

   @Override
   public void getCourseName() {
       System.out.println("Course Name: " + courseName);
   }

   @Override
   public void getCourseCode() {
       System.out.println("Course Code: " + courseCode);
   }

   @Override
   public void getAvailableSeats() {
       System.out.println("Course Available Seats: " + availableSeats);

   }

   @Override
   public void enrollCourse() {
       if (availableSeats > 0) {

           availableSeats--;
           System.out.println("✅| " + courseCode + " | " + courseName + " |
Seats : " + availableSeats + "/" + totalSeats + " | Enrolled!");
       } else {
           System.out.println("⚠| " + courseCode + " | " + courseName + " |
Seats : " + availableSeats + "/" + totalSeats + " | Full!");

       }
   }

   public void dropCourse() {
       if (availableSeats != totalSeats) {

           availableSeats++;
           System.out.println("❌| " + courseCode + " | " + courseName + " |
Seats : " + availableSeats + "/" + totalSeats + " | Dropped!");
       } else {
```

```
            System.out.println("⚠| " + courseCode + " | " + courseName + " |
Seats : " + availableSeats + "/" + totalSeats + " | No Enrollments!");

        }
    }
}
```

Interfaces:

```
package Task_4.Interfaces;

public interface CourseInfo {

    default void getCourseName() {
    }

    default void getCourseCode() {

    }

    default void getAvailableSeats() {
    }

}
```

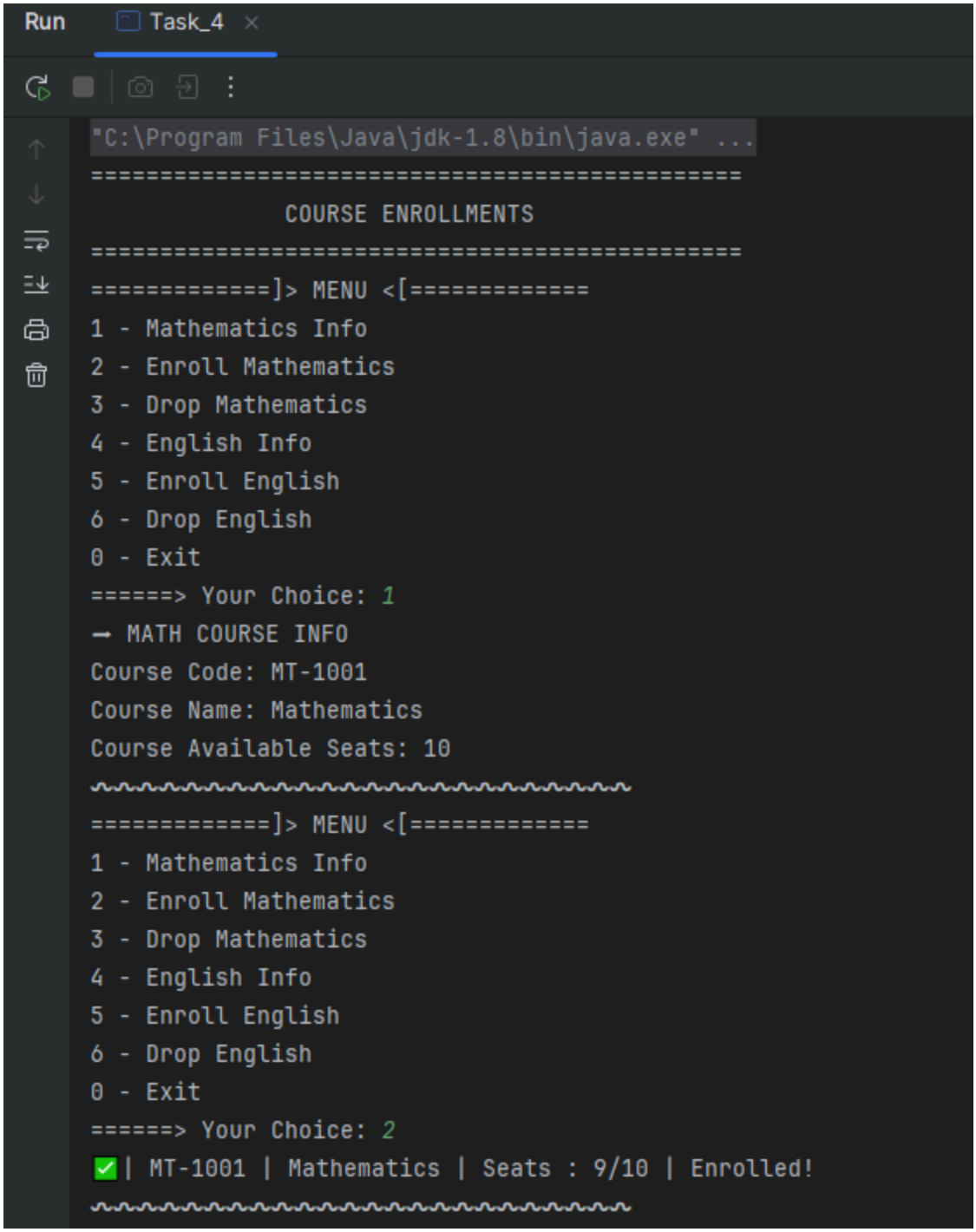
```
package Task_4.Interfaces;

public interface Enrollable {
    default void enrollCourse() {
    }

    default void dropCourse() {
    }
}
```

```
package Task_4;

import Task_4.Classes.English;
import Task_4.Classes.Maths;

import java.util.Scanner;

public class Task_4 {
    public static void main(String[] args) {
        try {

            Maths mathCourse = new Maths("MT-1001", "Mathematics", 10);
```

```
            English englishCourse = new English("EN-2002", "English", 10);

System.out.println("================================================");
            System.out.println("              COURSE ENROLLMENTS
");

System.out.println("================================================");
            while (true) {
                int selection;
                System.out.println("=============]> MENU <[=============");
                System.out.println("1 - Mathematics Info");
                System.out.println("2 - Enroll Mathematics");
                System.out.println("3 - Drop Mathematics");
                System.out.println("4 - English Info");
                System.out.println("5 - Enroll English");
                System.out.println("6 - Drop English");
                System.out.println("0 - Exit");
                System.out.print("======> Your Choice: ");
                Scanner scanner;
                scanner = new Scanner(System.in);
                selection = scanner.nextInt();

                switch (selection) {
                    case 1: {
                        System.out.println("➡ MATH COURSE INFO");
                        mathCourse.getCourseCode();
                        mathCourse.getCourseName();
                        mathCourse.getAvailableSeats();
                    }
                    break;
                    case 2: {
                        mathCourse.enrollCourse();
                    }
                    break;
                    case 3: {
                        mathCourse.dropCourse();
                    }
                    break;
                    case 4: {
                        System.out.println("➡ ENGLISH COURSE INFO");
                        englishCourse.getCourseCode();
                        englishCourse.getCourseName();
                        englishCourse.getAvailableSeats();
                    }
                    break;
                    case 5: {
                        englishCourse.enrollCourse();
                    }
                    break;
```

```
                    case 6: {
                        englishCourse.dropCourse();
                    }
                    break;
                    case 0: {
                        return;
                    }

                    default: {
                        System.out.println("❌Invalid Choice!");

                    }

                }
                System.out.println("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~");
            }


        } catch (Exception e) {
            System.out.println(e.toString());
        }
    }
}
```

Output:


Run
Task_4
"C:\Program Files\Java\jdk-1.8\bin\java.exe" ...
==========
COURSE ENROLLMENTS
==========
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 1
➡ MATH COURSE INFO
Course Code: MT-1001
Course Name: Mathematics
Course Available Seats: 10
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 2
✅| MT-1001 | Mathematics | Seats : 9/10 | Enrolled!

```
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 2
✅| MT-1001 | Mathematics | Seats : 8/10 | Enrolled!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 3
❌| MT-1001 | Mathematics | Seats : 9/10 | Dropped!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 4
```

```
======> Your Choice: 4
➡ ENGLISH COURSE INFO
Course Code: EN-2002
Course Name: English
Course Available Seats: 10
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 5
✅| EN-2002 | English | Seats : 9/10 | Enrolled!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 5
✅| EN-2002 | English | Seats : 8/10 | Enrolled!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 6
❌| EN-2002 | English | Seats : 9/10 | Dropped!
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
=============]> MENU <[=============
1 - Mathematics Info
2 - Enroll Mathematics
3 - Drop Mathematics
4 - English Info
5 - Enroll English
6 - Drop English
0 - Exit
======> Your Choice: 0

Process finished with exit code 0
```

Thanks