



# ULAB

---

UNIVERSITY OF LIBERAL ARTS  
BANGLADESH

**Lab Task: 1**

**Course code:** CSE2104

**Course Title:** Object Oriented Programming

**Section:** 2

**Semester:** Summer 2023

**Submitted to:** Shakib Mahmud Dipto

**Submitted by:**

Name: Jannatul Tazree

ID: 223014100

**Problem 1 : Create a class called BankAccount with instance variables accountNumber and balance. Add methods to deposit and withdraw money from the account .Create objects of BankAccount and perform deposit and withdrawal operations.**

**Answer :**

```
public class BankAccount {
    private String accountNumber;
    private double balance;

    public BankAccount(String accountNumber, double balance) {
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount + " taka");
        } else {
            System.out.println("Failed to deposit: Amount should be greater than zero.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0) {
            if (balance >= amount) {
                balance -= amount;
                System.out.println("Withdrew: " + amount + " taka");
            } else {
                System.out.println("Failed to withdraw: Insufficient balance.");
            }
        } else {
            System.out.println("Failed to withdraw: Invalid amount entered.");
        }
    }

    public double getBalance() {
        return balance;
    }
}
```

```

public void display() {
    System.out.println("Account Number: " + accountNumber);
    System.out.println("Balance: " + balance + " taka");
}

public static void main(String[] args) {
    BankAccount account1 = new BankAccount("223014100", 1000.0);
    account1.display();
    account1.deposit(500.0);
    account1.withdraw(200.0);
    account1.withdraw(2000.0);
    account1.display();
    BankAccount account2 = new BankAccount("223014098", 500.0);
    account2.display();
    account2.deposit(100.0);
    account2.withdraw(50.0);
    account2.display();
}
}

```

## Output:

The screenshot shows an IDE with a Java file named BankAccount.java. The code defines a BankAccount class with attributes accountNumber and balance, and methods for deposit, withdraw, and getBalance. The main method creates two BankAccount objects and performs various transactions on them. The output window shows the execution results, including account numbers, balances, and transaction details.

```

J BankAccount.java > Language Support for Java(TM) by Red Hat > BankAccount > main(String[])
1 public class BankAccount {
2     private String accountNumber;
3     private double balance;
4
5     public BankAccount(String accountNumber, double balance) {
6         this.accountNumber = accountNumber;
7         this.balance = balance;
8     }
9
10    public void deposit(double amount) {
11        if (amount > 0) {
12            balance += amount;
13            System.out.println("Deposited: " + amount + " taka");
14        } else {
15            System.out.println(x:"Failed to deposit: Amount should be greater than zero.");
16        }
17    }
18
19    public void withdraw(double amount) {
20        if (amount > 0) {
21            if (balance >= amount) {
22                balance -= amount;
23                System.out.println("Withdraw: " + amount + " taka");
24            } else {
25                System.out.println(x:"Failed to withdraw: Insufficient balance.");
26            }
27        } else {
28            System.out.println(x:"Failed to withdraw: Invalid amount entered.");
29        }
30    }
31
32    public double getBalance() {
33        return balance;
34    }
35 }

```

PROBLEMS 27 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Account Number: 223014100
Balance: 1000.0 taka
Deposited: 500.0 taka
Withdraw: 200.0 taka
Failed to withdraw: Insufficient balance.
Account Number: 223014100
Balance: 1300.0 taka
Account Number: 223014098
Balance: 500.0 taka
Deposited: 100.0 taka
Withdraw: 50.0 taka
Account Number: 223014098
Balance: 550.0 taka

```

**Problem 2:- Create a class rectangle with properties such as length and width. Add methods to calculate the perimeter and area of the rectangle. Create objects and display their corresponding perimeter and area.**

**Answer:**

```
public class Main {

    static class Rectangle {

        public int length;
        public int width;

        public Rectangle(int length, int width) {
            this.length = length;
            this.width = width;
        }

        public int calculatePerimeter() {
            return 2 * (length + width);
        }

        public int calculateArea() {
            return length * width;
        }

        public void displayDetails() {
            System.out.println("Length: " + length);
            System.out.println("Width: " + width);
            System.out.println("Perimeter: " + calculatePerimeter());
            System.out.println("Area: " + calculateArea());
            System.out.println();
        }
    }

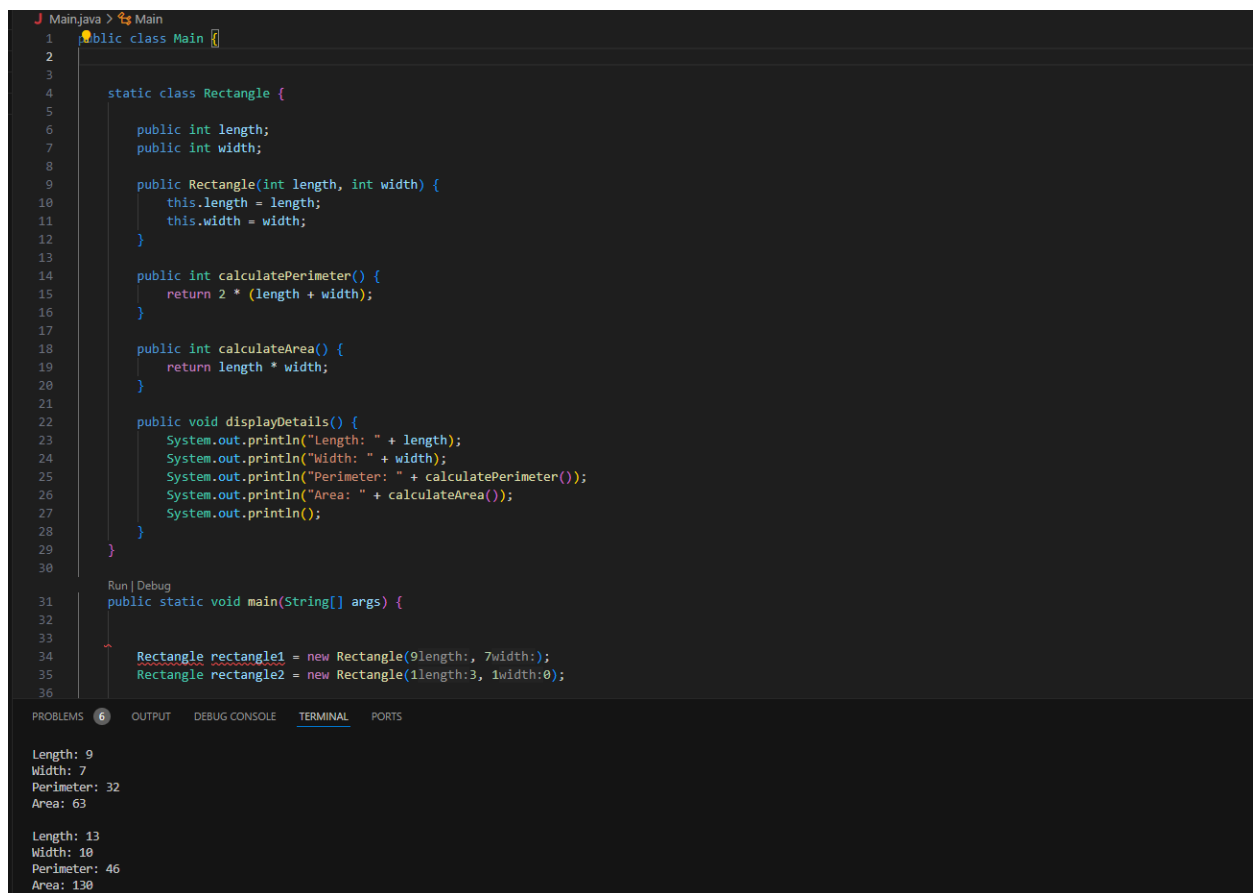
    public static void main(String[] args) {

        Rectangle rectangle1 = new Rectangle(9, 7);
```

```
Rectangle rectangle2 = new Rectangle(13, 10);

rectangle1.displayDetails();
rectangle2.displayDetails();
}
}
```

## Output:



```
J Main.java > Main
1 public class Main {
2
3
4     static class Rectangle {
5
6         public int length;
7         public int width;
8
9         public Rectangle(int length, int width) {
10             this.length = length;
11             this.width = width;
12         }
13
14         public int calculatePerimeter() {
15             return 2 * (length + width);
16         }
17
18         public int calculateArea() {
19             return length * width;
20         }
21
22         public void displayDetails() {
23             System.out.println("Length: " + length);
24             System.out.println("Width: " + width);
25             System.out.println("Perimeter: " + calculatePerimeter());
26             System.out.println("Area: " + calculateArea());
27             System.out.println();
28         }
29     }
30
31     Run | Debug
32     public static void main(String[] args) {
33
34         Rectangle rectangle1 = new Rectangle(9length: 9, 7width: 7);
35         Rectangle rectangle2 = new Rectangle(1length: 13, 1width: 10);
36     }
37 }
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Length: 9
Width: 7
Perimeter: 32
Area: 63

Length: 13
Width: 10
Perimeter: 46
Area: 130
```

**Problem 3: 3. Create a class called movie which as properties such as title, genre, lead actor, director, release year, rating and review. Create two movie objects and display their properties. If the rating is <5, the review should be “Not Good”. Otherwise, the review would be “Good”.**

**Answer :**

```
class Movie {

    private String title;
    private String genre;
    private String leadActor;
    private String director;
    private int releaseYear;
    private double rating;
    private String review;

    public Movie(String title, String genre, String leadActor, String director, int
releaseYear, double rating) {
        this.title = title;
        this.genre = genre;
        this.leadActor = leadActor;
        this.director = director;
        this.releaseYear = releaseYear;
        this.rating = rating;
        updateReview();
    }

    public void updateReview() {
        if (rating >= 5) {
            review = "Good";
        } else {
            review = "Not Good";
        }
    }

    public void displayDetails() {
        System.out.println("Title: " + title);
        System.out.println("Genre: " + genre);
        System.out.println("Lead Actor: " + leadActor);
        System.out.println("Director: " + director);
    }
}
```

```

        System.out.println("Release Year: " + releaseYear);
        System.out.println("Rating: " + rating);
        System.out.println("Review: " + review);
        System.out.println();
    }
}

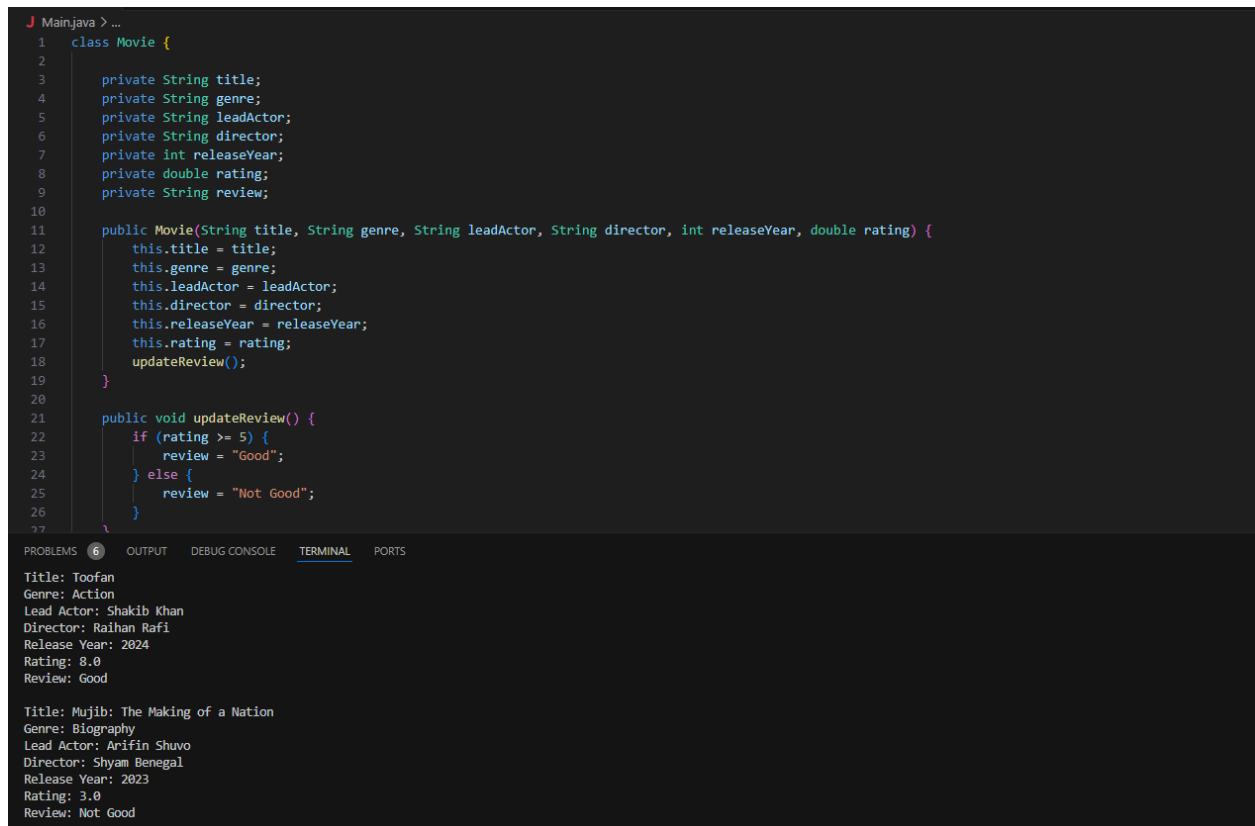
public class Main {
    public static void main(String[] args) {

        Movie movie1 = new Movie("Toofan", "Action", "Shakib Khan", "Raihan Rafi", 2024,
8.0);
        Movie movie2 = new Movie("Mujib: The Making of a Nation", "Biography", "Arifin
Shuvo", "Shyam Benegal", 2023, 3.0);

        movie1.displayDetails();
        movie2.displayDetails();
    }
}

```

## Output:



The screenshot shows an IDE with a Java file named 'Main.java'. The code defines a 'Movie' class with attributes for title, genre, lead actor, director, release year, rating, and review. It includes a constructor and an 'updateReview()' method that sets the review to 'Good' if the rating is 5 or higher, and 'Not Good' otherwise. The 'Main' class creates two 'Movie' objects: 'Toofan' (Action, 8.0 rating) and 'Mujib: The Making of a Nation' (Biography, 3.0 rating). The output window shows the details of these movies as printed by the 'displayDetails()' method.

```

J Main.java > ...
1  class Movie {
2
3      private String title;
4      private String genre;
5      private String leadActor;
6      private String director;
7      private int releaseYear;
8      private double rating;
9      private String review;
10
11     public Movie(String title, String genre, String leadActor, String director, int releaseYear, double rating) {
12         this.title = title;
13         this.genre = genre;
14         this.leadActor = leadActor;
15         this.director = director;
16         this.releaseYear = releaseYear;
17         this.rating = rating;
18         updateReview();
19     }
20
21     public void updateReview() {
22         if (rating >= 5) {
23             review = "Good";
24         } else {
25             review = "Not Good";
26         }
27     }
28 }

```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Title: Toofan
Genre: Action
Lead Actor: Shakib Khan
Director: Raihan Rafi
Release Year: 2024
Rating: 8.0
Review: Good

Title: Mujib: The Making of a Nation
Genre: Biography
Lead Actor: Arifin Shuvo
Director: Shyam Benegal
Release Year: 2023
Rating: 3.0
Review: Not Good

```

