

```
//1
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
 license
 */
```

```
package com.mycompany.mamunproject1;
```

```
/**
 *
 * @author Rasel Mamun
 */
public class Mamunproject1 {

    public static void main(String[] args) {
        System.out.println("MD Rasel Mamun");
        System.out.println ("2223413126");
    }
}
```

```
//2
// code solution 2
import java.util.Scanner;

public class NewClass1 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Simple Calculator");

        System.out.print("Enter first number: ");

        double num1 = scanner.nextDouble();

        System.out.print("Enter second number: ");

        double num2 = scanner.nextDouble();

        System.out.println("Choose operation: +, -, *, /");

        char operation = scanner.next().charAt(0);
```

```

double result = 0;
boolean validOperation = true;

switch (operation) {
    case '+':
        result = num1 + num2;
        break;
    case '-':
        result = num1 - num2;
        break;
    case '*':
        result = num1 * num2;
        break;
    case '/':
        if (num2 != 0) {
            result = num1 / num2;
        } else {
            System.out.println("Error! Division by zero is not allowed.");
            validOperation = false;
        }
        break;
    default:
        System.out.println("Error! Invalid operation.");
        validOperation = false;
        break;
}

if (validOperation) {
    System.out.println("Result: " + result);
}

scanner.close();
}
}

```

//3

// 3

```

public class MaxMin {

    public static void main(String[] args) {

        int[] numbers = {3, 5, 7, 2, 8, -1, 4, 10, 12,15};
    }
}

```

```

int max = numbers[0];
int min = numbers[0];

for (int i = 1; i < numbers.length; i++) {
    if (numbers[i] > max) {
        max = numbers[i];
    }
    if (numbers[i] < min) {
        min = numbers[i];
    }
}

System.out.println("Max value: " + max);
System.out.println("Min value: " + min);
}
}

```

```

//4
//code solution 4
public class Rasel {

    public static void main(String[] args) {

        int size = 8;

        for (int i = 0; i < size; i++) {

            for (int j = 0; j < size; j++) {

                if ((i + j) % 2 == 0) {
                    System.out.print("1 ");
                } else {
                    System.out.print("0 ");
                }
            }
        }

        System.out.println();
    }
}

```

```
    }  
  }  
}
```

//5

//Code solution 5

```
import java.util.Scanner;
```

```
public class Sum {
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        System.out.print("Enter a positive integer: ");  
        int n = scanner.nextInt();
```

```
        int sum = 0;
```

```
        for (int i = 1; i <= n; i++) {  
            sum += i;  
        }
```

```
        System.out.println("The sum of integers from 1 to " + n + " is: " + sum);
```

```
        scanner.close();  
    }  
}
```

//6

//7 code solution

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

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

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

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

    public String getAccountNumber() {
        return accountNumber;
    }

    public double getBalance() {
        return balance;
    }

    public void setAccountNumber(String accountNumber) {
        this.accountNumber = accountNumber;
    }

    public void setBalance(double balance) {
        this.balance = balance;
    }

    public void deposit(double amount) {
        if (amount > 0) {
            balance += amount;
            System.out.println("Deposited: " + amount);
        } else {
            System.out.println("Deposit amount must be positive.");
        }
    }

    public void withdraw(double amount) {
        if (amount > 0 && amount <= balance) {
            balance -= amount;
            System.out.println("Withdrew: " + amount);
        } else {
            System.out.println("Invalid withdraw amount.");
        }
    }
}
```

```

    }
}

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

```

// Customer class

```

class Customer {
    private String name;
    private String customerID;
    private List<BankAccount> bankAccounts;

    public Customer(String name, String customerID) {
        this.name = name;
        this.customerID = customerID;
        this.bankAccounts = new ArrayList<>();
    }

    public String getName() {
        return name;
    }

    public String getCustomerID() {
        return customerID;
    }

    public void setName(String name) {
        this.name = name;
    }

    public void setCustomerID(String customerID) {
        this.customerID = customerID;
    }

    public void addBankAccount(BankAccount account) {
        bankAccounts.add(account);
    }
}

```

```
public void displayCustomerInfo() {  
    System.out.println("Customer Name: " + name);  
    System.out.println("Customer ID: " + customerID);  
    System.out.println("Accounts:");  
    for (BankAccount account : bankAccounts) {  
        account.displayAccountInfo();  
    }  
}
```

```
public void deposit(String accountNumber, double amount) {  
    for (BankAccount account : bankAccounts) {  
        if (account.getAccountNumber().equals(accountNumber)) {  
            account.deposit(amount);  
            return;  
        }  
    }  
    System.out.println("Account not found.");  
}
```

```
public void withdraw(String accountNumber, double amount) {  
    for (BankAccount account : bankAccounts) {  
        if (account.getAccountNumber().equals(accountNumber)) {  
            account.withdraw(amount);  
            return;  
        }  
    }  
    System.out.println("Account not found.");  
}  
}
```