

DAY 2
IF-EISE

```
class Account{
    public static void main(String[] args){
        double balance = 600;
        System.out.println("Amount to withdraw");
        double amount = 500;

        if (amount <= 0 || amount > balance){
            System.out.println("Withdrawal has failed");
        }else{
            balance-=amount;
            System.out.println("Withdrawal has succeeded");
        }
    }
}
```

Output:

Output

```
java -cp /tmp/rrQuzF5nAT Account
Amount to withdrawWithdrawal has succeeded
```

2] If else if else-

```
class Account{
    public static void main(String[] args){
        double balance = 600;
        System.out.println("Amount to withdraw");
        double amount = 500;
        if(amount<=0){
            System.out.println("Withdrawal has failed as the amount is negative");
        }
        else if(amount>balance){
            System.out.println("Withdrawal has failed as the balance is low");
        }
        else{
            balance-=amount;
            System.out.println("Withdrawal has succeeded");
        }
    }
}
```

```
}  
}  
}
```

Output

```
java -cp /tmp/rrQuzF5nAT Account  
Amount to withdraw  
Withdrawal has succeeded  
|
```

3] Switch Case

```
class Account {  
    public static void main(String[] args) {  
        int choice = 0;  
        choice = 2; // Input taken from user  
        switch(choice) { // choice passed to switch statement  
            // choice has to match the case value to execute the statements in that case option.  
            case 1: // new record entry: newEntry()  
                System.out.println("Entry deposition");  
                break;  
  
            case 2: // displaying details of account: display()  
                System.out.println("Display operation");  
                break; // break keyword is used to exit from the switch block.  
  
            case 3: // deposit operation: deposit()  
                System.out.println("Deposit operation");  
                break;  
  
            default: // If no case matches, default will be executed.  
                System.out.println("Invalid choice");  
        }  
    }  
}
```

Output

```
java -cp /tmp/rrQuzF5nAT Account  
Display operation
```

4] do-while loop:

```
import java.util.Scanner;  
class Account {  
    public static void main(String[] args) {  
        double balance = 0;  
        double minbal = 500;  
        double depositAmt = 0;  
        //Scanner sc = new Scanner(System.in); uncomment when working in eclipse  
  
        do {  
            System.out.println("$100 have been added to the account");  
            depositAmt +=100;           // harcode different depositAmt values  
            //depositAmt = sc.nextInt();    uncomment when working in eclipse  
        } while(depositAmt < minbal);  
        balance = depositAmt;  
        System.out.println("Transaction Complete");  
    }  
}
```

Output

```
java -cp /tmp/rrQuzF5nAT Account  
$100 have been added to the account  
$100 have been added to the account  
$100 have been added to the account  
$100 have been added to the account  
$100 have been added to the account  
Transaction Complete  
|
```

5] while loop

```
import java.util.Scanner;  
class Account {  
    public static void main(String[] args) {  
        double balance = 0;  
        double minbal = 500;  
        double depositAmt = 0;  
        //Scanner sc = new Scanner(System.in);    // uncomment when working in eclipse  
        while(depositAmt < minbal) {  
            System.out.println("$100 have been added to account");  
            //depositAmt = sc.nextInt();           // harcode different depositAmt values  
                                                    // uncomment when working in eclipse  
            depositAmt +=100;  
        }  
        balance = depositAmt;  
        System.out.println("Transaction Complete");  
    }  
}
```

Output

```
java -cp /tmp/rrQuzF5nAT Account  
$100 have been added to account  
$100 have been added to account  
$100 have been added to account  
$100 have been added to account  
$100 have been added to account  
Transaction Complete
```

6] for loop

```
class Account {  
    public static void main(String[] args) {  
        double balance = 6000, rateOfInterest = 0.10, interest = 0;  
        double withdrawal = 500, deposit = 600;  
        for(int i = 1; i <= 12; ++i) {  
            balance += deposit;  
            balance -= withdrawal;  
            interest = balance * rateOfInterest;  
            balance += interest;  
            System.out.println("The interest for the month " + i + " is " + interest);  
        }  
        System.out.println("The balance at the end of the year is " + balance);  
    }  
}
```

Output

```
java -cp /tmp/rrQuzF5nAT Account
The interest for the month 1 is 600.0
The interest for the month 2 is 660.0
The interest for the month 3 is 726.0
The interest for the month 4 is 798.6
The interest for the month 5 is 878.46
The interest for the month 6 is 966.3060000000002
The interest for the month 7 is 1062.9366000000002
The interest for the month 8 is 1169.2302600000003
The interest for the month 9 is 1286.1532860000004
The interest for the month 10 is 1414.7686146000005
The interest for the month 11 is 1556.2454760600006
The interest for the month 12 is 1711.8700236660006
The balance at the end of the year is 18830.570260326007
|
```

7]

One dimensional primitive array

```
class Bank {
    public static void main(String[] args) {
        int[] phone = new int[3]; // primitive type array
        phone[0] = 7120686;
        phone[1] = 7120687; // inserting element in an array
        phone[2] = 7120684;
        int count = 1;
        for(int i = 0; i < phone.length; ++i) { // traversing an array
            System.out.println("phone number " + count + ": " + phone[i]);
            ++count;
        }
    }
}
```

Output

```
java -cp /tmp/rrQuzF5nAT Bank
phone number 1: 7120686
phone number 2: 7120687
phone number 3: 7120684
|
```

8] Reference type arrays

```
class bank{
    public static void main(String[] args){
        Customer[] customer = new Customer[2]; //Reference type Array
        Customer customer1 = new Customer("Anil", "Acc12345");
        Customer customer2 = new Customer("Ajay", "Acc12346");
        customer[0] = customer1; //storing in the array
        customer[1] = customer2;
        for(int i=0;i<customer.length;i++){ //traversing the array
            Customer customeObject = customer[i]; //retrieving customer Object
            String name = customeObject.displayCustomerName();
            System.out.println("the customer name is..." + name);
        }
    }
}

class Customer{
    private String name;
    private String customerId;

    Customer(String unname, String ucustomerId){
        name = unname;
        customerId = ucustomerId;
    }

    public String displayCustomerName(){
        return name;
    }
}
```

```
}
```

Output

```
java -cp /tmp/rrQuzF5nAT bank  
the customer name is...Anilthe customer name is...Ajay  
|
```

9] Enhanced for loop

```
class Account{  
    public static void main(String[] args){  
        double balance = 6000;  
        double rateOfInterest = 0.10;  
        double interest = 0;  
        double withdrawal = 500;  
        double deposit = 600;  
        int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};  
        for(int i: arr) { // The iteration in the loop happens automatically. The value is assigned  
to  
            //variable i from the array in every iteration of the loop.  
            balance += deposit; // Loop will repeat the statements in its body till the last element  
is reached in the array.  
            balance -= withdrawal;  
            interest = balance * rateOfInterest;  
            balance += interest;  
            System.out.println("The interest for the month " + i + " is " + interest);  
        }  
    }  
}
```


Output

Clear

```
java -cp /tmp/rrQuzF5nAT Acccount
The interest for the month 1 is 610.0
The interest for the month 2 is 681.0
The interest for the month 3 is 759.1
The interest for the month 4 is 845.0100000000001
The interest for the month 5 is 939.5110000000001
The interest for the month 6 is 1043.4621000000002
The interest for the month 7 is 1157.8083100000001
The interest for the month 8 is 1283.5891410000004
The interest for the month 9 is 1421.9480551000004The interest for the month 10 is
1574.1428606100003
The interest for the month 11 is 1741.5571466710005
The interest for the month 12 is 1925.7128613381003
```

10] constructors

```
class Bank {
    private String bankName, area;
    private String phoneNumber;
    Bank() { // Default constructor
        bankName = "IBank";
        area = "Gandhi Nagar";
        phoneNumber = "9876543210";
    }
    Bank(String bname, String barea, String phoneNo) { // Parameterized constructor
        bankName = bname;
        area = barea;
        phoneNumber = phoneNo;
    }
    void displayBankDetails(){
        System.out.println("bank Name: " + bankName);
        System.out.println("Area of bank: " + area);
        System.out.println("Phone number of bank: " + phoneNumber);
    }

    public static void main(String[] args){
        Bank bank1 = new Bank(); //call default constructor
        Bank bank2 = new Bank("IBank", "Jaydev Nagar", "8876543219"); //call Parameterized
        constructor

        bank1.displayBankDetails();
        System.out.println("*****");
    }
}
```

```

        bank2.displayBankDetails();
    }
}

```

Output

```

java -cp /tmp/rrQuzF5nAT Bank
bank Name: IBank
Area of bank: Gandhi Nagar
Phone number of bank: 9876543210
*****
bank Name: IBank
Area of bank: Jaydev Nagar
Phone number of bank: 8876543219
|

```

11] This keyword

```

class Bank {
    private String bankName; //instance variable
    private String area;
    private String phoneNumber;

    Bank(String bankName, String area, String phoneNumber) { // Parameterized constructor

        this.bankName = bankName; //this keyword is used to assign
        this.area = area;          //the value for instance variables
        this.phoneNumber = phoneNumber;

    }

    void displayBankDetails(){
        System.out.println("bank Name: " + bankName);
        System.out.println("Area of bank: " + area);
        System.out.println("Phone number of bank: " + phoneNumber);

    }

    public static void main(String[] args){

```

```

        Bank bank = new Bank("IBank", "Jaydev Nagar", "8876543210"); //call default
        constructor
        bank.displayBankDetails();

    }
}

```

Output

```

java -cp /tmp/rrQuzF5nAT Bank
bank Name: IBankArea of bank: Jaydev Nagar
Phone number of bank: 8876543210|

```

12] inheritance

```

class Loan {
    int tenure;
    double principal;
    float interestRate;
    String accountNumber;
    public double calculateEMI(){
        double simpleInterest = (principal*interestRate*tenure)/100;
        return (simpleInterest+principal)/tenure;
    }
}

class HomeLoan extends Loan {
    HomeLoan() {
        tenure = 5; //reusing super class member variables
        principal = 20000;
        interestRate = 8.5f;
        accountNumber = "Acc12345";
    }

    public static void main(String[] args) {
        HomeLoan hloan = new HomeLoan();
        double amount = hloan.calculateEMI(); // sub class Object
        // invoking super class method
        System.out.println("emi per year..." + amount);
    }
}

```

```
}
```

Output

```
java -cp /tmp/rrQuzF5nAT HomeLoan  
emi per year...5700.0
```

13] Method Overloading

```
class Loan{  
    private float interest;  
  
    Loan(){  
        interest = 8.5f;  
    }  
  
    //calculateEMI overloaded methods  
    public double calculateEMI(int tenure, double principal){  
        double simpleInterest = (principal * interest * tenure) / 100;  
        return (simpleInterest+principal)/tenure;  
    }  
    public double calculateEMI(double principal, int tenure){  
        double simpleInterest = (principal * interest * tenure) / 100;  
        return (simpleInterest+principal)/tenure;  
    }  
    public double calculateEMI(int tenure, double principal, float interest){  
        double simpleInterest = (principal * interest * tenure) / 100;  
        return (simpleInterest+principal)/tenure;  
    }  
  
    public static void main(String[] args){  
        Loan loan = new Loan();  
        double result = loan.calculateEMI(20000d, 5); //d means double  
        double value = loan.calculateEMI(5, 20000d);  
        double val = loan.calculateEMI(5, 20000, 9.5f); // f means float  
  
        System.out.println("EMI per year is..." + result);  
        System.out.println("EMI per year is..." + value);  
    }  
}
```

```

        System.out.println("EMI per year is..." + val);
    }
}

```

Output

```

java -cp /tmp/rrQuzF5nAT Loan
EMI per year is...5700.0
EMI per year is...5700.0
EMI per year is...5900.0|

```

14] Method overriding

```

class Loan{
    public double calculateEMI(double principal){
        double simpleInterest = (principal * 8.5 * 5) / 100;
        return (simpleInterest+principal)/5;
    }
}

class HomeLoan extends Loan {
    // method overridden
    public double calculateEMI(double principal){
        int additionalTax = 200;
        double simpleInterest = (principal * 7.5 * 20) / 100;
        double emi = (simpleInterest + principal) / 20;
        return emi+additionalTax;
    }
}

```

```

class VehicleLoan extends Loan {
    // method overridden
    public double calculateEMI(double principal){
        int additionalTax = 200;
        double simpleInterest = (principal * 9.5 * 10) / 100;
        double emi = (simpleInterest + principal) / 10;
    }
}

```

```
        return emi+additionalTax;
    }
}
```

```
class ExecuteLoan{
    public static void main(String[] args){
        Loan loan = null;
        loan = new HomeLoan();    //Runtime Polymorphism
        double hloan = loan.calculateEMI(2000000);
        loan = new VehicleLoan(); //sup class reference holding sub class Object
        double vloan = loan.calculateEMI(100000);
        System.out.println("Home loan emi per year is..." + hloan);
        System.out.println("Vehicle loan emi per year is..." + vloan);

    }
}
```

Output

Clear

```
java -cp /tmp/rrQuzF5nAT ExecuteLoan
```

```
Home loan emi per year is...250200.0Vehicle loan emi per year is...19700.0
```