Experiment:3

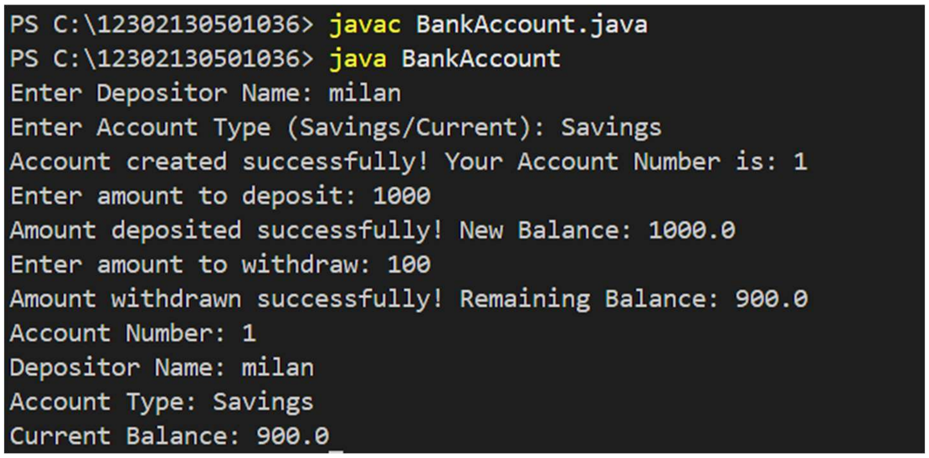
1.Create a class BankAccount that has Depositor name , Acc_no, Acc_type, Balance as Data Members and void createAcc() . void Deposit(), void withdraw() and void BalanceInquiry as Member Function. When a new Account is created assign next serial no as account number. Account number starts from 1

Solution :

```
import java.util.Scanner;
class BankAccount {
   private static int nextAccNo = 1;
   private String depositorName;
   private int accNo;
   private String accType;
   private double balance;

   public BankAccount() {
      this.accNo = nextAccNo++;
      this.balance = 0.0;
   }

   public void createAcc() {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter Depositor Name: ");
      this.depositorName = scanner.nextLine();
      System.out.print("Enter Account Type (Savings/Current): ");
      this.accType = scanner.nextLine();
      System.out.println("Account created successfully! Your Account Number is: " +
this.accNo);
   }
```

```
    public void deposit() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter amount to deposit: ");
        double amount = scanner.nextDouble();
        if (amount > 0) {
            this.balance += amount;
            System.out.println("Amount deposited successfully! New Balance: " +
this.balance);
        } else {
            System.out.println("Invalid amount. Please enter a positive value.");
        }
    }

    public void withdraw() {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter amount to withdraw: ");
        double amount = scanner.nextDouble();
        if (amount > 0 && amount <= this.balance) {
            this.balance -= amount;
            System.out.println("Amount withdrawn successfully! Remaining Balance: " +
this.balance);
        } else {
            System.out.println("Invalid transaction. Insufficient balance or invalid amount.");
        }
    }

    public void balanceInquiry() {
        System.out.println("Account Number: " + this.accNo);
        System.out.println("Depositor Name: " + this.depositorName);
        System.out.println("Account Type: " + this.accType);
        System.out.println("Current Balance: " + this.balance);
    }

    public static void main(String[] args) {
        BankAccount account = new BankAccount();
        account.createAcc();
        account.deposit();
        account.withdraw();
        account.balanceInquiry();
    }
}
```

Output:



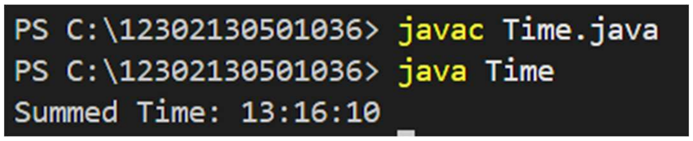
```
PS C:\12302130501036> javac BankAccount.java
PS C:\12302130501036> java BankAccount
Enter Depositor Name: milan
Enter Account Type (Savings/Current): Savings
Account created successfully! Your Account Number is: 1
Enter amount to deposit: 1000
Amount deposited successfully! New Balance: 1000.0
Enter amount to withdraw: 100
Amount withdrawn successfully! Remaining Balance: 900.0
Account Number: 1
Depositor Name: milan
Account Type: Savings
Current Balance: 900.0
```

2. Create a class time that has hour, minute and second as data members. Create a parameterized constructor to initialize Time Objects. Create a member Function Time Sum (Time, Time) to sum two time objects.

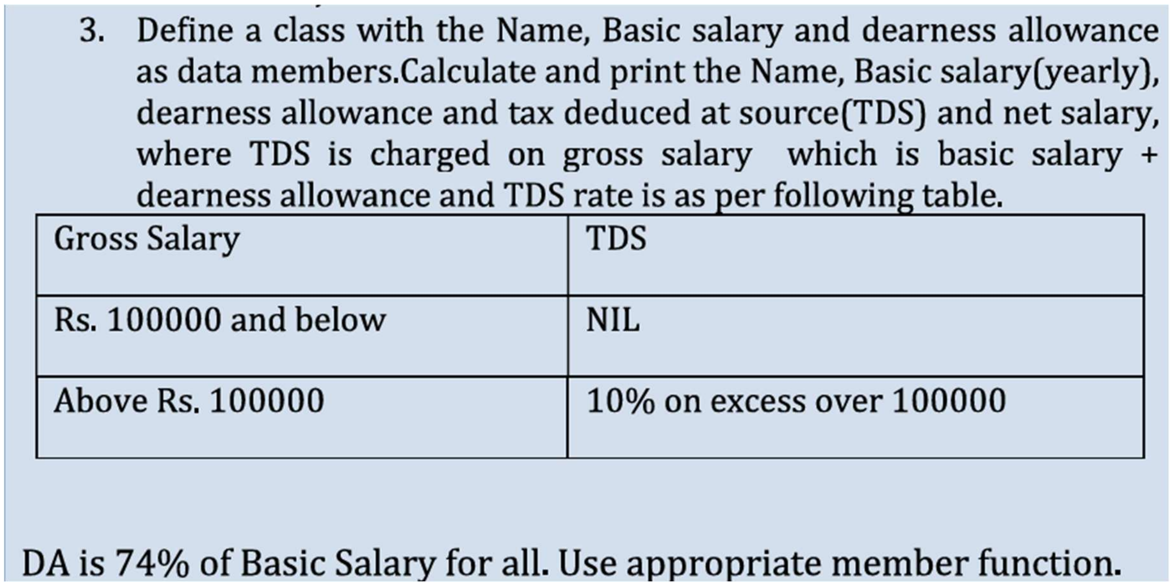
Solution :

```
class Time {
    private int hour;
    private int minute;
    private int second;
    public Time(int hour, int minute, int second) {
        this.hour = hour;
        this.minute = minute;
        this.second = second;
    }

    public static Time sum(Time t1, Time t2) {
        int totalSeconds = t1.second + t2.second;
        int extraMinutes = totalSeconds / 60;
        int finalSecond = totalSeconds % 60;
        int totalMinutes = t1.minute + t2.minute + extraMinutes;
        int extraHours = totalMinutes / 60;
        int finalMinute = totalMinutes % 60;
        int finalHour = (t1.hour + t2.hour + extraHours) % 24; // Ensures hour remains
within 24-hour format
        return new Time(finalHour, finalMinute, finalSecond);
    }
    public void displayTime() {
        System.out.printf("%02d:%02d:%02d\n", hour, minute, second);
    }

    public static void main(String[] args) {
        Time t1 = new Time(10, 45, 50);
        Time t2 = new Time(2, 30, 20);
        Time sumTime = Time.sum(t1, t2);
        System.out.print("Summed Time: ");
        sumTime.displayTime();
    }
}
```

Output:



```
PS C:\12302130501036> javac Time.java
PS C:\12302130501036> java Time
Summed Time: 13:16:10
```



3. Define a class with the Name, Basic salary and dearness allowance as data members.Calculate and print the Name, Basic salary(yearly), dearness allowance and tax deduced at source(TDS) and net salary, where TDS is charged on gross salary which is basic salary + dearness allowance and TDS rate is as per following table.

Gross Salary	TDS
Rs. 100000 and below	NIL
Above Rs. 100000	10% on excess over 100000

DA is 74% of Basic Salary for all. Use appropriate member function.

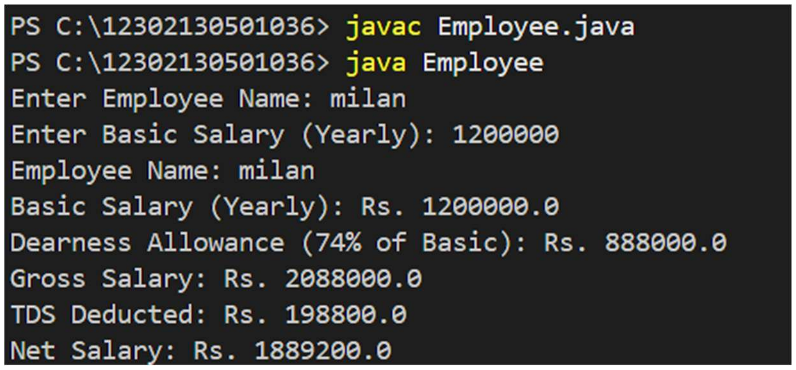
Solution :

```
import java.util.Scanner;
class Employee {
   private String name;
   private double basicSalary;
   private double dearnessAllowance;

   public Employee(String name, double basicSalary) {
      this.name = name;
      this.basicSalary = basicSalary;
      this.dearnessAllowance = basicSalary * 0.74;
   }

   public double getGrossSalary() {
      return basicSalary + dearnessAllowance;
   }
```

```
public double calculateTDS() {
    double grossSalary = getGrossSalary();
    if (grossSalary > 100000) {
        return (grossSalary - 100000) * 0.10;
    }
    return 0;
}

public double getNetSalary() {
    return getGrossSalary() - calculateTDS();
}

public void displayDetails() {
    System.out.println("Employee Name: " + name);
    System.out.println("Basic Salary (Yearly): Rs. " + basicSalary);
    System.out.println("Dearness Allowance (74% of Basic): Rs. " + dearnessAllowance);
    System.out.println("Gross Salary: Rs. " + getGrossSalary());
    System.out.println("TDS Deducted: Rs. " + calculateTDS());
    System.out.println("Net Salary: Rs. " + getNetSalary());
}

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.print("Enter Employee Name: ");
    String name = scanner.nextLine();
    System.out.print("Enter Basic Salary (Yearly): ");
    double basicSalary = scanner.nextDouble();
    Employee emp = new Employee(name, basicSalary);
```

```
        emp.displayDetails();
    }
}
```

Output:



```
PS C:\12302130501036> javac Employee.java
PS C:\12302130501036> java Employee
Enter Employee Name: milan
Enter Basic Salary (Yearly): 1200000
Employee Name: milan
Basic Salary (Yearly): Rs. 1200000.0
Dearness Allowance (74% of Basic): Rs. 888000.0
Gross Salary: Rs. 2088000.0
TDS Deducted: Rs. 198800.0
Net Salary: Rs. 1889200.0
```