Ex. No. 4	CREATING USER DEFINED DATATYPES USING CLASSES AND OBJECTS
Date of Exercise	10-08-23

1.**Aim:**

To write a menu driven application to perform the ATM operations using JAVA. Your application must contain the following functionalities. Use constructors, getter and setter functions..

Procedure:

Step1:start the program
Step2:set bank balance
Step3: use getter and setter
Step4: in switch case we create menu driven
Step5:stop the program

Program:

```
import java.util.Scanner;
class ATM {
    private int accountNumber;
    private int pin;
    private double balance;
    public ATM(int accountNumber, int pin, double balance) {
        this.accountNumber = accountNumber;
        this.pin = pin;
        this.balance = balance;
    }
    public int getAccountNumber() {
        return accountNumber;
    }
    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }
```

```
public int getPin() {
    return pin;
}
public void setPin(int pin) {
    this.pin = pin;
}
public double getBalance() {
    return balance;
}
public void setBalance(double balance) {
    this.balance = balance;
}
public void balanceEnquiry() {
    System.out.println("Current Balance: $" + balance);
}
public void withdraw(double amount) {
    if (balance >= amount && amount > 0 && amount % 100 == 0) {
        balance -= amount;
        System.out.println("Withdrawal successful. Remaining Balance: $" + balance);
    } else if (balance < amount) {
        System.out.println("Insufficient balance.");
    } else {
        System.out.println("Invalid withdrawal amount.");
    }
}
public void changePIN(int newPin) {
    pin = newPin;
    System.out.println("PIN changed successfully.");
```

```
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ATM atm = new ATM(12345, 1234, 1000);
        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Balance Enquiry");
            System.out.println("2. Withdrawal");
            System.out.println("3. Change PIN");
            System.out.println("4. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            switch (choice) {
                case 1:
                    atm.balanceEnquiry();
                    break;
                case 2:
                    System.out.print("Enter withdrawal amount: $");
                    double amount = scanner.nextDouble();
                    atm.withdraw(amount);
                    break;
                case 3:
                    System.out.print("Enter new PIN: ");
                    int newPin = scanner.nextInt();
                    atm.changePIN(newPin);
                    break;
```

```
            case 4:
                System.out.println("Exiting the application.");
                scanner.close();
                System.exit(0);
                break;
            default:
                System.out.println("Invalid choice. Please select a valid option.");
        }
    }
  }
}
```

Output:

```
Menu:
1. Balance Enquiry
2. Withdrawal
3. Change PIN
4. Exit
Enter your choice: 1
Current Balance: $1000.0

Menu:
1. Balance Enquiry
2. Withdrawal
3. Change PIN
4. Exit
Enter your choice: 2
Enter withdrawal amount: $500
Withdrawal successful. Remaining Balance: $500.0

Menu:
1. Balance Enquiry
2. Withdrawal
3. Change PIN
4. Exit
Enter your choice: 3
Enter new PIN: 1606
PIN changed successfully.

Menu:
1. Balance Enquiry
2. Withdrawal
3. Change PIN
4. Exit
Enter your choice: 4
Exiting the application.
```

Result:

The above program has been successfully executed and verified.

2.**Aim:**
Create a class called time that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. Another member function should display it, in 11:59:59 format. The final member function should add two objects of type time passed as arguments. A main () program should create two initialized time objects and one that isn't initialized.

Procedure:
Step1:start the program
Step2: get time from user
Step3: add two objects of type time passed
Step4: print added time
Step5:stop the program

Program:

```
import java.util.Scanner;
class Time {
   private int hrs, mins, secs;
   public Time() {
      hrs = 0;
      mins = 0;
      secs = 0;
   }
   public Time(int h, int m, int s) {
      hrs = h;
      mins = m;
      secs = s;
   }
   public void display() {
      System.out.println(hrs + ":" + mins + ":" + secs);
   }
   public void addTime(Time t1, Time t2) {
      secs = t1.secs + t2.secs;
      if (secs > 59) {
```

```
            secs -= 60;
            mins++;
        }
        mins += t1.mins + t2.mins;
        if (mins > 59) {
            mins -= 60;
            hrs++;
        }
        hrs += t1.hrs + t2.hrs;
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter hours for t1: ");
        int hrs1 = scanner.nextInt();
        System.out.print("Enter minutes for t1: ");
        int mins1 = scanner.nextInt();
        System.out.print("Enter seconds for t1: ");
        int secs1 = scanner.nextInt();
        Time t1 = new Time(hrs1, mins1, secs1);
        System.out.print("Enter hours for t2: ");
        int hrs2 = scanner.nextInt();
        System.out.print("Enter minutes for t2: ");
        int mins2 = scanner.nextInt();
        System.out.print("Enter seconds for t2: ");
        int secs2 = scanner.nextInt();
        Time t2 = new Time(hrs2, mins2, secs2);
        Time t3 = new Time();
```

```
        t3.addTime(t1, t2);
        System.out.println("Sum of: ");
        t1.display();
        t2.display();
        System.out.print("is: ");
        t3.display();
        scanner.close();
    }
}
```

Output:

```
Enter hours for t1: 2
Enter minutes for t1: 2
Enter seconds for t1: 2
Enter hours for t2: 11
Enter minutes for t2: 11
Enter seconds for t2: 11
Sum of:
2:2:2
11:11:11
is: 13:13:13


...Program finished with exit code 0
Press ENTER to exit console.
```

Result:

The above program has been successfully executed and verified.

3.**Aim:**
Write a menu driven application to maintain the student information using JAVA. Your application must have the following functionalities. Use constructors, getter and setter functions. name, registerNo, department, cgpa,

Procedure:
Step1:start the program
Step2: create the class
Step3: use getter and setter
Step4: get student details from user
Step5: print student details
Step6:stop the program

Program:

```
import java.util.Scanner;
class Student {
   private String name;
   private String registerNo;
   private String department;
   private double cgpa;
   public Student(String name, String registerNo, String department, double cgpa) {
      this.name = name;
      this.registerNo = registerNo;
      this.department = department;
      this.cgpa = cgpa;
   }
   public String getName() {
      return name;
   }
   public String getRegisterNo() {
      return registerNo;
   }
   public String getDepartment() {
```

```
        return department;
    }
    public double getCgpa() {
        return cgpa;
    }
}
public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Student[] students = new Student[100];
        int studentCount = 0;
        while (true) {
            System.out.println("\nMenu:");
            System.out.println("1. Add Student");
            System.out.println("2. Display Students");
            System.out.println("3. Exit");
            System.out.print("Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline
            switch (choice) {
                case 1:
                    if (studentCount < students.length) {
                        System.out.print("Enter Name: ");
                        String name = scanner.nextLine();
                        System.out.print("Enter Register No: ");
                        String registerNo = scanner.nextLine();
                        System.out.print("Enter Department: ");
                        String department = scanner.nextLine();
                        System.out.print("Enter CGPA: ");
```

```
                double cgpa = scanner.nextDouble();
                scanner.nextLine(); // Consume newline

                students[studentCount] = new Student(name, registerNo, department, cgpa);
                studentCount++;
                System.out.println("Student added successfully.");
            } else {
                System.out.println("Maximum student limit reached.");
            }
            break;
        case 2:
            System.out.println("\nStudent Details:");
            for (int i = 0; i < studentCount; i++) {
                Student student = students[i];
                System.out.println("Name: " + student.getName());
                System.out.println("Register No: " + student.getRegisterNo());
                System.out.println("Department: " + student.getDepartment());
                System.out.println("CGPA: " + student.getCgpa());
                System.out.println("---------------------"); }
            break;
        case 3:
            System.out.println("Exiting...");
            System.exit(0);
        default:
            System.out.println("Invalid choice. Please try again.");
        }
    }
}
}
```

Output:

```
Menu:
1. Add Student
2. Display Students
3. Exit
Enter your choice: 1
Enter Name: hari
Enter Register No: 48
Enter Department: ai
Enter CGPA: 9
Student added successfully.

Menu:
1. Add Student
2. Display Students
3. Exit
Enter your choice: 2

Student Details:
Name: hariharan K
Register No: 1048
Department: cse AI DS
CGPA: 9.99
----------------------
Name: hari
Register No: 48
Department: ai
CGPA: 9.0
----------------------

Menu:
1. Add Student
2. Display Students
3. Exit
Enter your choice: 3
Exiting...
```

Result:

The above program has been successfully executed and verified.