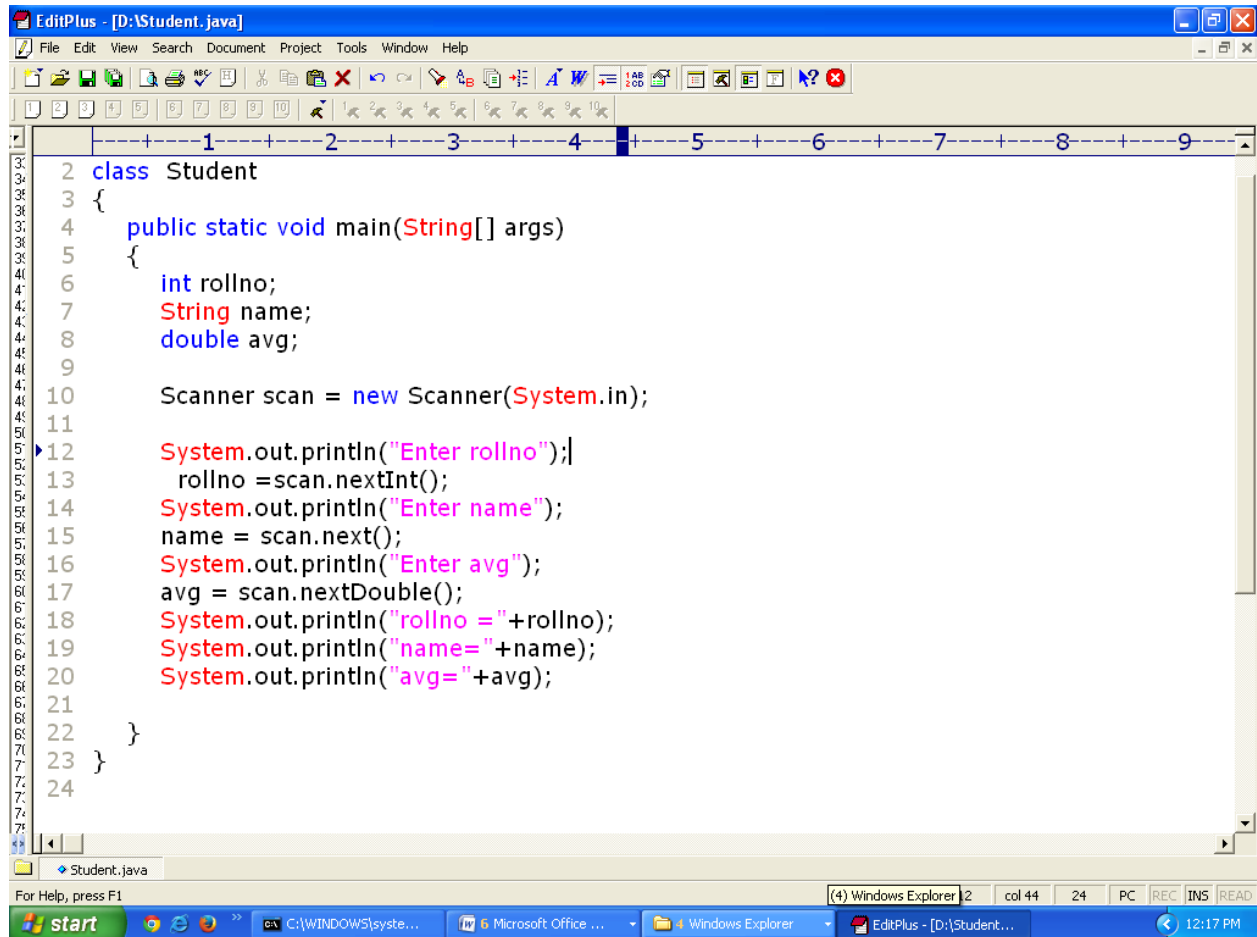


Lab-2 Assignments

Introducing Scanner Class To Accept The Values From The User

Demo for Scanner Class

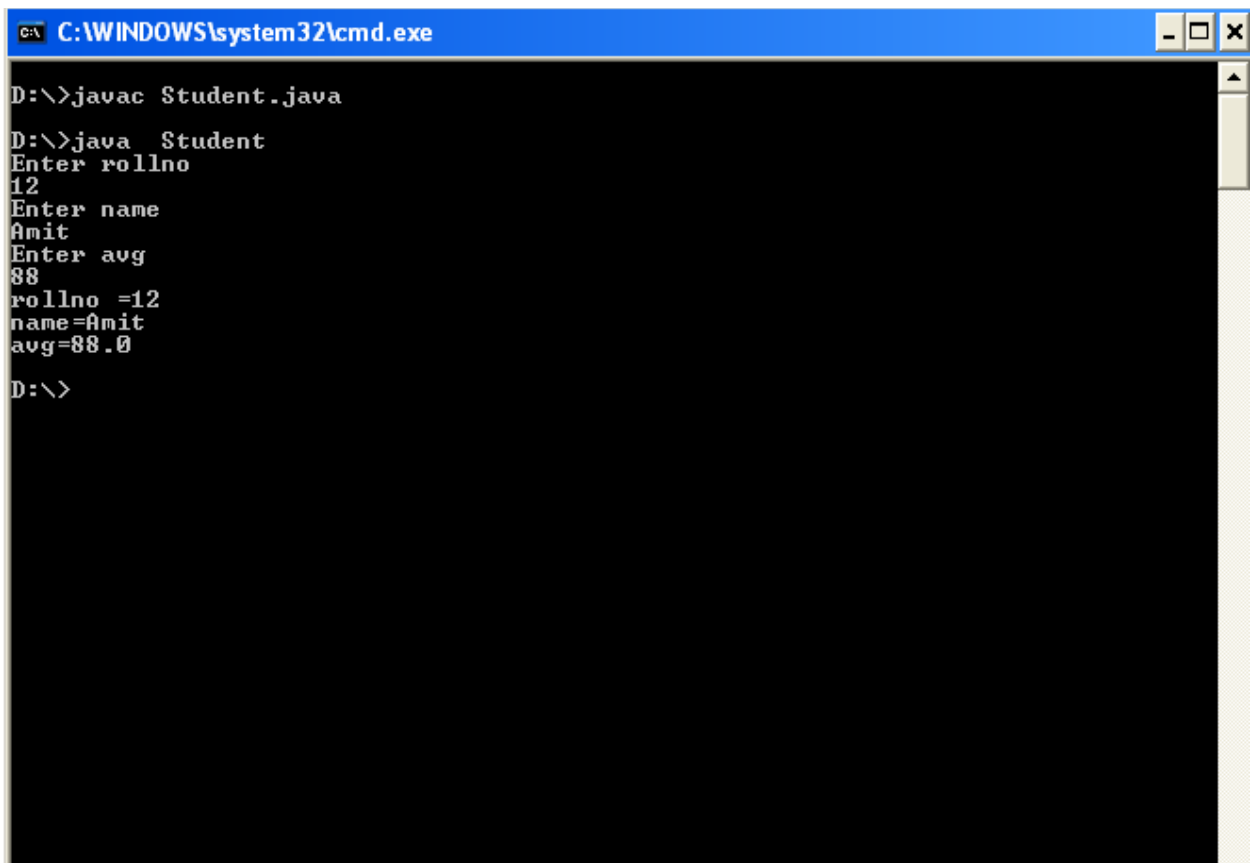


```
1  class Student
2  {
3      public static void main(String[] args)
4      {
5          int rollno;
6          String name;
7          double avg;
8
9          Scanner scan = new Scanner(System.in);
10
11          System.out.println("Enter rollno");
12          rollno = scan.nextInt();
13          System.out.println("Enter name");
14          name = scan.next();
15          System.out.println("Enter avg");
16          avg = scan.nextDouble();
17          System.out.println("rollno = "+rollno);
18          System.out.println("name = "+name);
19          System.out.println("avg = "+avg);
20
21      }
22  }
```

Save as Student.java(in d:\)

Compile and run from Command Prompt

Note: Always set the path if new command prompt is opened(as mentioned in the previous lab demo)



A screenshot of a Windows command prompt window. The title bar at the top is blue and contains the text "C:\WINDOWS\system32\cmd.exe" along with standard window control buttons (minimize, maximize, close). The command prompt itself has a black background with white text. The text shows the following sequence of commands and output:

```
D:\>javac Student.java
D:\>java Student
Enter rollno
12
Enter name
Amit
Enter avg
88
rollno =12
name=Amit
avg=88.0
D:\>
```

Demo for Getter Setter Methods

The screenshot shows the EditPlus IDE with a Java file named 'calldate.java' open. The code defines a 'date' class with the following methods: 'initDate', 'setDate', 'setMonth', 'setYear', 'getDate', 'getMonth', 'getYear', and 'toString'. The 'toString' method returns the date in 'dd/mm/yyyy' format. The IDE interface includes a menu bar, a toolbar, a line number margin, and a status bar at the bottom showing 'Ln 11, col 38, 53'.

```
1 class date{
2     int date,month,year;
3     public void initDate(int d, int m, int y){
4         date =d;
5         month=m;
6         year =y;
7     }
8     public void setDate(int date){
9         this.date=date;
10    }
11    public void setMonth(int month){
12        this.month=month;
13    }
14    public void setYear(int year){
15        this.year=year;
16    }
17    public int getDate(){
18        return date;
19    }
20    public int getMonth(){
21        return month;
22    }
23    public int getYear(){
24        return year;
25    }
26
27    public String toString() {
28        return date+"/"+month+"/"+year;
29    }
30 }
31
```

ac1

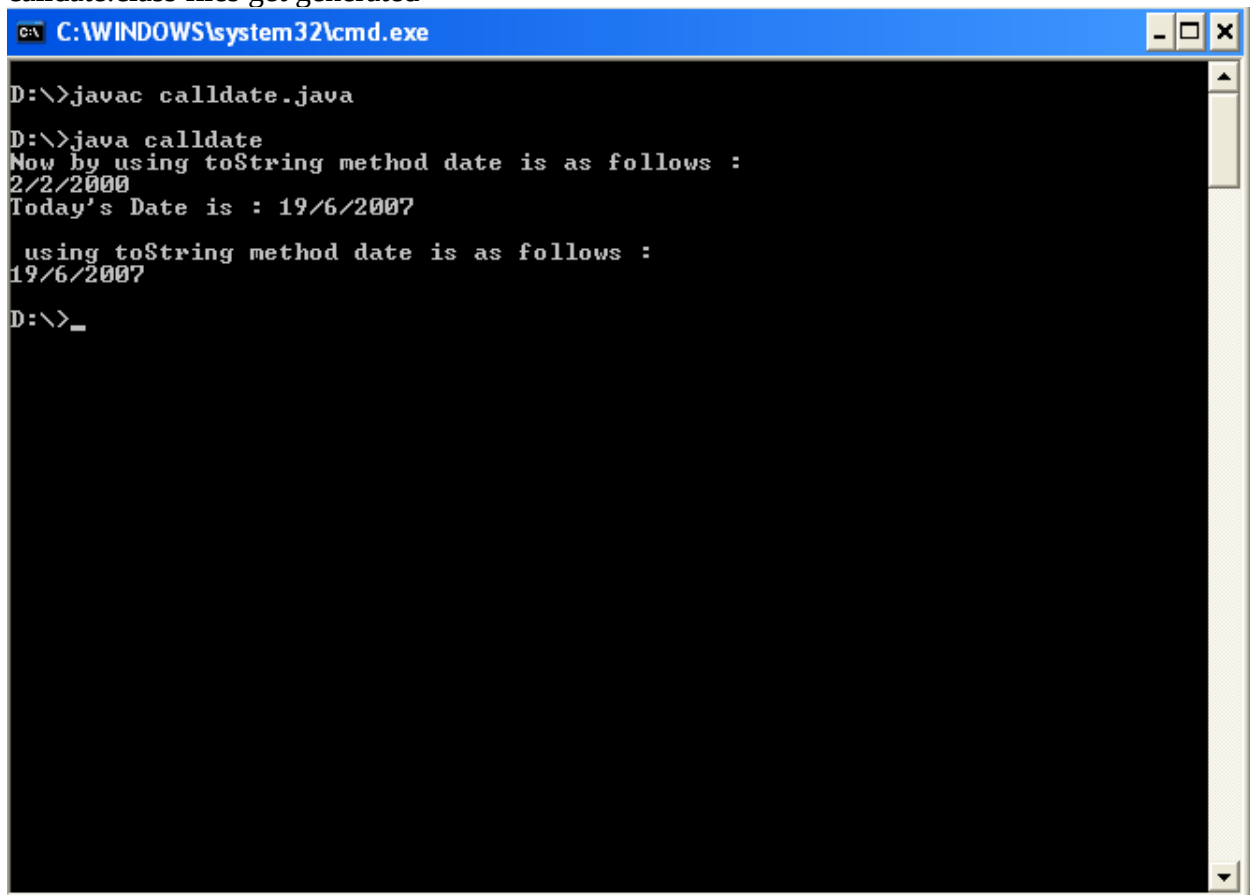
The screenshot shows the EditPlus IDE with the same 'calldate.java' file. The code now includes a 'main' method that creates a 'date' object, initializes it, and demonstrates the 'set' and 'toString' methods. The IDE interface is consistent with the previous screenshot, but the status bar at the bottom shows 'Ln 76, col 28, 80'.

```
60
61 class calldate
62 {
63     public static void main(String[] args)
64     {
65         date d=new date();
66         d.initDate(2,2,2000);
67         System.out.println("Now by using toString method date is as follows : ");
68         System.out.println(d);
69         d.setDate(19);
70         d.setMonth(06);
71         d.setYear(2007);
72
73         System.out.println("Today's Date is : "+d.getDate()+"/"+d.getMonth()+"/"+d.getYear());
74
75         System.out.println();
76         System.out.println("using toString method date is as follows : ");
77         System.out.println(d);
78     }
79 }
80
```

Save as calldate.java

Compile and run calldate.java

Note: When calldate.java is compiled, date class also gets compiled and date.class and calldate.class files get generated



```
C:\WINDOWS\system32\cmd.exe

D:\>javac calldate.java

D:\>java calldate
Now by using toString method date is as follows :
2/2/2000
Today's Date is : 19/6/2007

    using toString method date is as follows :
19/6/2007
D:\>_
```

Assignments To Solve

1. WAP to check whether a person is eligible for voting
2. WAP to check whether a given year is a LEAP year
3. WAP to find the factorial of a number?
4. WAP to find whether a given number is an Armstrong number
5. WAP to reverse the given number
6. Generate the Fib... series 1 1 2 3 5 8 13
7. Create a class Book which describes its Book_title and Book_price. Use getter and setter methods to get & set the Books description. Implement createBook and showBook methods to create and display Book Information. Write a separate class BookInfo to access the Book class