

Name: Ammar Khan

Reg No: FA24-BSE-148

Course: OOPs

Class: 3C

LAB 05

Exercise: CAR

```
Source History
1 package ammar1;
2
3 import java.util.Scanner;
4
5 public class Ammar1 {
6     private String name;
7     private char direction;
8     private int position;
9
10    public Ammar1(String name, char direction, int position) {
11        this.name = name;
12        this.direction = Character.toUpperCase(direction);
13        this.position = position;
14    }
15
16    public void turn() {
17        switch (direction) {
18            case 'N':
19                direction = 'E';
20                break;
21            case 'E':
22                direction = 'S';
23                break;
24            case 'S':
25                direction = 'W';
26                break;
27            case 'W':
28                direction = 'N';
29                break;
30            default:
31                System.out.println("Invalid direction!");
32        }
33    }
34
35    public void turn(char newDirection) {
36        newDirection = Character.toUpperCase(newDirection);
37        if (newDirection == 'N' || newDirection == 'E' || newDirection == 'S' || newDirection == 'W')
38            direction = newDirection;
39        else {
40            System.out.println("Invalid direction! Use N, E, S or W.");
41        }
42    }
43 }
```

```

43
44 public void move(int distance) {
45     switch (direction) {
46         case 'N':
47         case 'E':
48             position += distance;
49             break;
50         case 'S':
51         case 'W':
52             position -= distance;
53             break;
54         default:
55             System.out.println("Invalid direction!");
56     }
57 }
58
59 public void display() {
60     System.out.println("Car Name: " + name);
61     System.out.println("Direction: " + direction);
62     System.out.println("Position: " + position);
63 }
64
65 public static void main(String[] args) {
66     Scanner sc = new Scanner(System.in);
67
68     // Input car name
69     System.out.print("Enter car name: ");
70     String name = sc.nextLine();
71
72     // Input direction with validation
73     char direction;
74     while (true) {
75         System.out.print("Enter initial direction (N, E, S, W): ");
76         String input = sc.nextLine().toUpperCase();
77         if (input.length() == 1 && "NESW".indexOf(input.charAt(0)) != -1) {
78             direction = input.charAt(0);
79             break;
80         } else {
81             System.out.println("Invalid direction! Please enter N, E, S, or W.");
82         }
83     }

```

```

85
86     System.out.print("Enter initial position (integer): ");
87     int position = sc.nextInt();
88
89     Ammar1 car = new Ammar1(name, direction, position);
90
91     car.display();
92
93     System.out.println("\nTurning right by one step:");
94     car.turn();
95     car.display();
96
97     System.out.println("\nTurning directly to West:");
98     car.turn('W');
99     car.display();
100
101     System.out.println("\nMoving 10 units:");
102     car.move(10);
103     car.display();
104
105     System.out.println("\nMoving 5 units:");
106     car.move(5);
107     car.display();
108
109     sc.close();
110 }
111

```

ammar1.Ammar1 > main >

Output - ammar1 (run) X

```

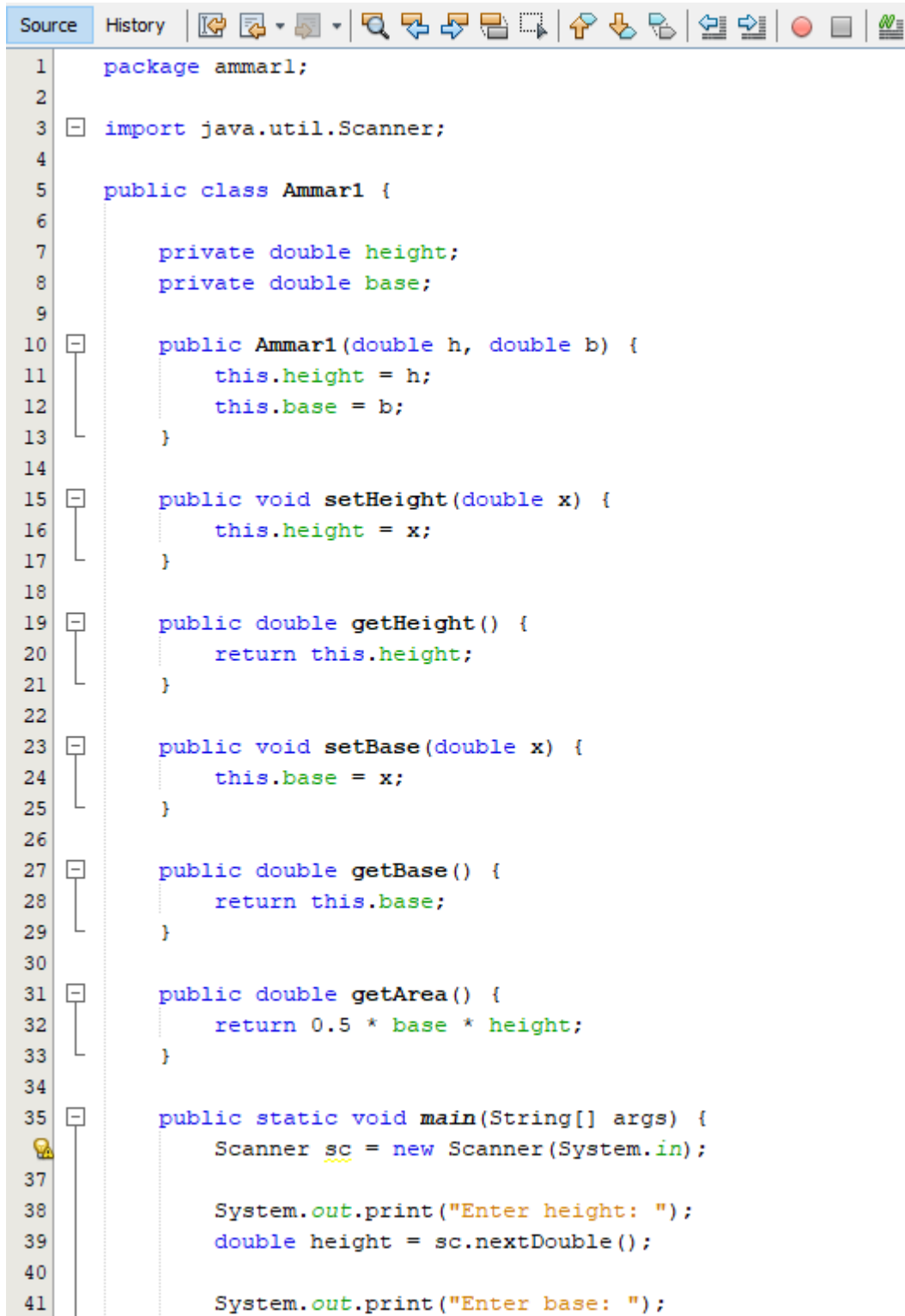
run:
Enter car name: suzuki
Enter initial direction (N, E, S, W): w
Enter initial position (integer): 2
Car Name: suzuki
Direction: W
Position: 2

Turning right by one step:
Car Name: suzuki
Direction: N
Position: 2

Turning directly to West:
Car Name: suzuki
Direction: W
Position: 2

```

Exercise: Triangle



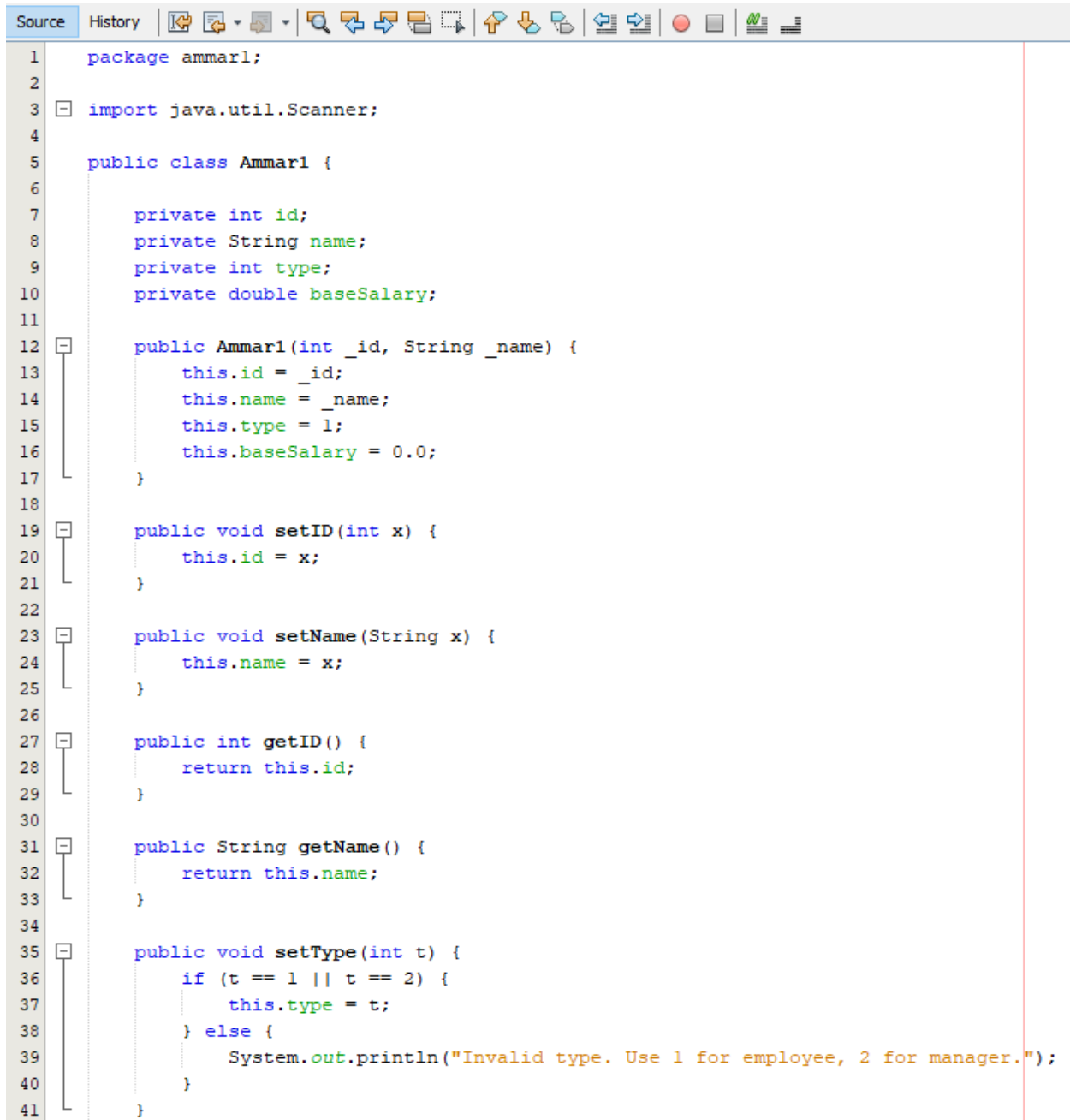
```
1 package ammar1;
2
3 import java.util.Scanner;
4
5 public class Ammar1 {
6
7     private double height;
8     private double base;
9
10    public Ammar1(double h, double b) {
11        this.height = h;
12        this.base = b;
13    }
14
15    public void setHeight(double x) {
16        this.height = x;
17    }
18
19    public double getHeight() {
20        return this.height;
21    }
22
23    public void setBase(double x) {
24        this.base = x;
25    }
26
27    public double getBase() {
28        return this.base;
29    }
30
31    public double getArea() {
32        return 0.5 * base * height;
33    }
34
35    public static void main(String[] args) {
36        Scanner sc = new Scanner(System.in);
37
38        System.out.print("Enter height: ");
39        double height = sc.nextDouble();
40
41        System.out.print("Enter base: ");
```

```
42         double base = sc.nextDouble();
43
44         Ammar1 t = new Ammar1(height, base);
45
46         System.out.println("Height: " + t.getHeight());
47         System.out.println("Base: " + t.getBase());
48         System.out.println("Area: " + t.getArea());
49
50         System.out.print("Enter updated height: ");
51         t.setHeight(sc.nextDouble());
52
53         System.out.print("Enter updated base: ");
54         t.setBase(sc.nextDouble());
55
56         System.out.println("Updated Height: " + t.getHeight());
57         System.out.println("Updated Base: " + t.getBase());
58         System.out.println("Updated Area: " + t.getArea());
59
60         sc.close();
61     }
62 }
```

Output - ammar1 (run) ×

```
run:
Enter height: 5.11
Enter base: 4
Height: 5.11
Base: 4.0
Area: 10.22
Enter updated height: 6
Enter updated base: 5
Updated Height: 6.0
Updated Base: 5.0
Updated Area: 15.0
BUILD SUCCESSFUL (total time: 23 seconds)
```

Exercise: Employee



The screenshot shows an IDE window with a toolbar at the top and a code editor below. The code editor contains the following Java code:

```
1 package ammar1;
2
3 import java.util.Scanner;
4
5 public class Ammar1 {
6
7     private int id;
8     private String name;
9     private int type;
10    private double baseSalary;
11
12    public Ammar1(int _id, String _name) {
13        this.id = _id;
14        this.name = _name;
15        this.type = 1;
16        this.baseSalary = 0.0;
17    }
18
19    public void setID(int x) {
20        this.id = x;
21    }
22
23    public void setName(String x) {
24        this.name = x;
25    }
26
27    public int getID() {
28        return this.id;
29    }
30
31    public String getName() {
32        return this.name;
33    }
34
35    public void setType(int t) {
36        if (t == 1 || t == 2) {
37            this.type = t;
38        } else {
39            System.out.println("Invalid type. Use 1 for employee, 2 for manager.");
40        }
41    }
42 }
```

```

42
43     public void setBaseSalary(double bs) {
44         this.baseSalary = bs;
45     }
46
47     public double getSalary() {
48         if (type == 2) {
49             return baseSalary * 1.10;
50         } else {
51             return baseSalary;
52         }
53     }
54
55     public static void main(String[] args) {
56         Scanner sc = new Scanner(System.in);
57
58         System.out.print("Enter employee ID: ");
59         int id = sc.nextInt();
60         sc.nextLine(); // consume newline
61
62         System.out.print("Enter employee name: ");
63         String name = sc.nextLine();
64
65         Ammar1 emp = new Ammar1(id, name);
66
67         System.out.print("Enter employee type (1 = employee, 2 = manager): ");
68         int type = sc.nextInt();
69         emp.setType(type);
70
71         System.out.print("Enter base salary: ");
72         double salary = sc.nextDouble();
73         emp.setBaseSalary(salary);
74
75         System.out.println(emp.getName() + " Salary: " + emp.getSalary());
76
77         sc.close();
78     }
79 }

```

Output - ammar1 (run) ×

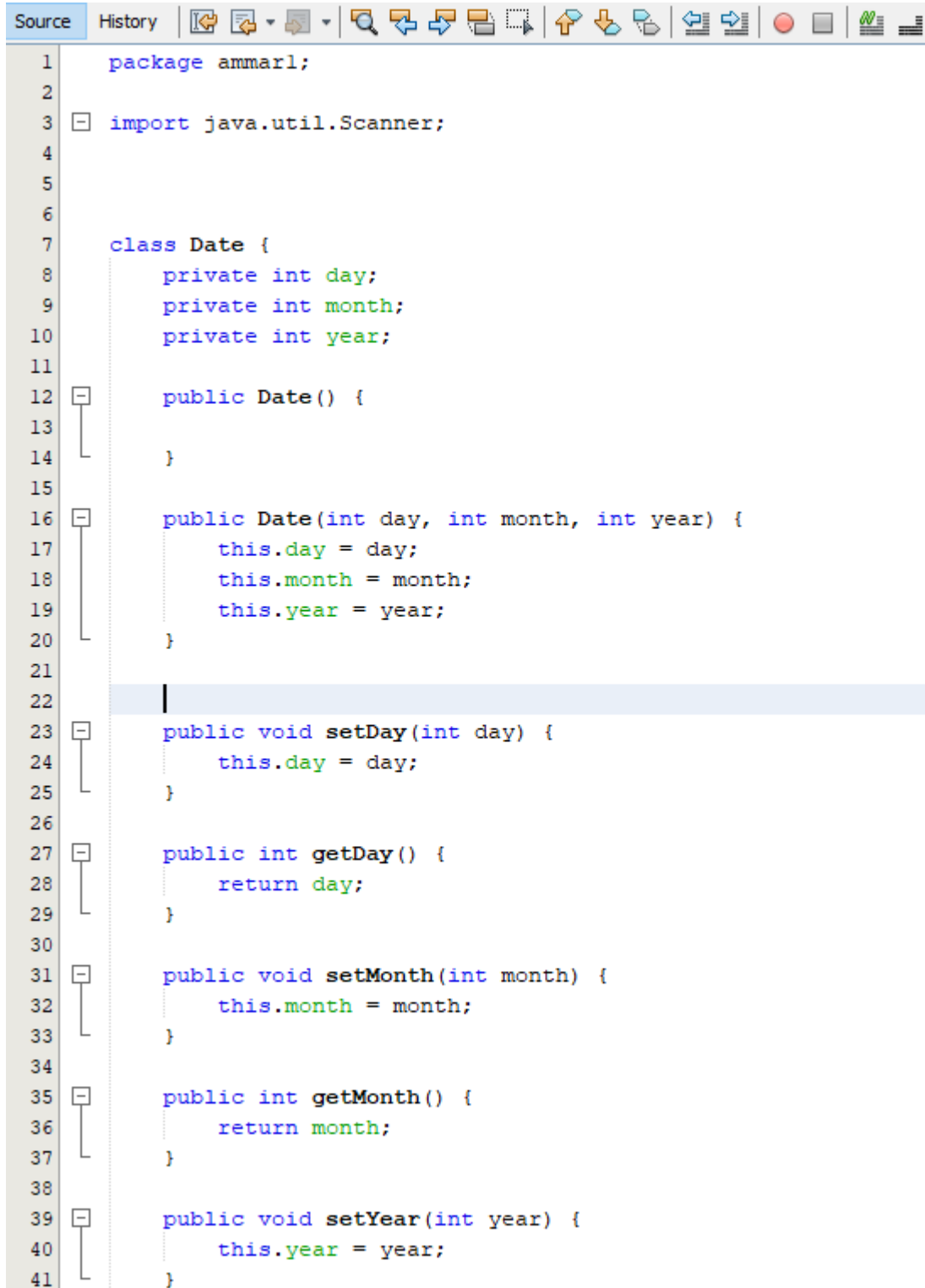


```

run:
Enter employee ID: 23
Enter employee name: ammar
Enter employee type (1 = employee, 2 = manager): 2
Enter base salary: 40088
ammar Salary: 44096.8
BUILD SUCCESSFUL (total time: 19 seconds)

```

Exercise: Date



The image shows a screenshot of a Java IDE with a source code editor. The code defines a `Date` class within the `ammar1` package. The class has three private integer fields: `day`, `month`, and `year`. It includes a no-argument constructor, a three-argument constructor, and setter/getter methods for each field. The IDE interface includes a toolbar with various icons for file operations, navigation, and execution, and a sidebar with tabs for 'Source' and 'History'.

```
1 package ammar1;
2
3 import java.util.Scanner;
4
5
6
7 class Date {
8     private int day;
9     private int month;
10    private int year;
11
12    public Date() {
13    }
14
15
16    public Date(int day, int month, int year) {
17        this.day = day;
18        this.month = month;
19        this.year = year;
20    }
21
22
23    public void setDay(int day) {
24        this.day = day;
25    }
26
27    public int getDay() {
28        return day;
29    }
30
31    public void setMonth(int month) {
32        this.month = month;
33    }
34
35    public int getMonth() {
36        return month;
37    }
38
39    public void setYear(int year) {
40        this.year = year;
41    }
```



```
42
43 public int getYear() {
44     return year;
45 }
46
47 @Override
48 public String toString() {
49     return day + "/" + month + "/" + year;
50 }
51 }
52
53 class Person {
54     private String name;
55     private int age;
56     private Date dob;
57
58     public Person() {
59
60     }
61
62     public Person(String name, int age, Date dob) {
63         this.name = name;
64         this.age = age;
65         this.dob = dob;
66     }
67
68     // getters and setters
69     public void setName(String name) {
70         this.name = name;
71     }
72
73     public String getName() {
74         return name;
75     }
76
77     public void setAge(int age) {
78         this.age = age;
79     }
80 }
```

```

81 public int getAge() {
82     return age;
83 }
84
85 public void setDob(Date dob) {
86     this.dob = dob;
87 }
88
89 public Date getDob() {
90     return dob;
91 }
92
93 public void display() {
94     System.out.println("Name: " + name);
95     System.out.println("Age: " + age);
96     System.out.println("Date of Birth: " + dob);
97 }
98 }
99
100 public class TestPersonDate {
101     public static void main(String[] args) {
102         Scanner sc = new Scanner(System.in);
103
104
105         System.out.println("Enter details for Person 1:");
106         System.out.print("Name: ");
107         String name1 = sc.nextLine();
108
109         System.out.print("Age: ");
110         int age1 = sc.nextInt();
111
112         System.out.println("Enter Date of Birth (day month year): ");
113         int day1 = sc.nextInt();
114         int month1 = sc.nextInt();
115         int year1 = sc.nextInt();
116         sc.nextLine();
117
118         Date dob1 = new Date(day1, month1, year1);
119         Person p1 = new Person(name1, age1, dob1);

```

```

121
122     System.out.println("\nEnter details for Person 2:");
123     System.out.print("Name: ");
124     String name2 = sc.nextLine();
125
126     System.out.print("Age: ");
127     int age2 = sc.nextInt();
128
129     System.out.println("Enter Date of Birth (day month year): ");
130     int day2 = sc.nextInt();
131     int month2 = sc.nextInt();
132     int year2 = sc.nextInt();
133
134     Date dob2 = new Date(day2, month2, year2);
135     Person p2 = new Person(name2, age2, dob2);
136
137
138     System.out.println("\nPerson 1 Details:");
139     p1.display();
140
141     System.out.println("\nPerson 2 Details:");
142     p2.display();
143
144     sc.close();
145 }
146

```

Output - ammar1 (run) ×

```

run:
Enter details for Person 1:
Name: ammar
Age: 19
Enter Date of Birth (day month year):
09 11 2005

Enter details for Person 2:
Name: hassan
Age: 22
Enter Date of Birth (day month year):
09
12
2003

Person 1 Details:
Name: ammar
Age: 19
Date of Birth: 9/11/2005

Person 2 Details:
Name: hassan
Age: 22
Date of Birth: 9/12/2003

```