

Assignment - 2

Vishwas Cp

Engineering Intern

Tecnotree Mysore

1. Create a class called "Car" that has the following properties: make, model, year, color, and price. Include a constructor and getter and setter methods for each property.

<https://codeshare.io/eV6xlb>

The screenshot shows a Java development environment with two tabs: 'Car.java' and 'Console'. The 'Car.java' tab displays the following code:

```
1 System.out.print("Enter car make: ");
2 String make = scanner.nextLine();
3
4 System.out.print("Enter car model: ");
5 String model = scanner.nextLine();
6
7 System.out.print("Enter car year: ");
8 int year = scanner.nextInt();
9 scanner.nextLine(); // consume newline left-over
10
11 System.out.print("Enter car color: ");
12 String color = scanner.nextLine();
13
14 System.out.print("Enter car price: ");
15 double price = scanner.nextDouble();
16
17 Car car = new Car(make, model, year, color, price);
18
19 System.out.println("Make: " + car.getMake());
20 System.out.println("Model: " + car.getModel());
21 System.out.println("Year: " + car.getYear());
22 System.out.println("Color: " + car.getColor());
```

The 'Console' tab shows the application's output after running:

```
<terminated> Car [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 3:39:46 pm – 3:41:26 pm) [pid: 21120]
Enter car make: BMW
Enter car model: BMW C500
Enter car year: 2019
Enter car color: Black
Enter car price: 500000
Make: BMW
Model: BMW C500
Year: 2019
Color: Black
Price: 500000.0
```

2. Create a class called "Student" that has the following properties: name, age, gender, grade, and GPA. Include a constructor and getter and setter methods for each property.

<https://codeshare.io/JbMKZX>

The screenshot shows an IDE interface with two tabs: 'Student.java' and 'Console'. The code in 'Student.java' defines a class 'Student' with private fields for name, age, gender, grade, and GPA, and a constructor that initializes these fields. The 'Console' tab shows the output of running the program, where it prompts for name, age, gender, grade, and GPA, and then prints them back out.

```
1 package VishwasCp/src/assignment2;
2 import java.util.Scanner;
3
4 public class Student {
5
6
7     private String name;
8     private int age;
9     private String gender;
10    private int grade;
11    private double GPA;
12
13    // Constructor
14    public Student(String name, int age, String gender, int grade, double GPA) {
15        this.name = name;
16        this.age = age;
17        this.gender = gender;
18        this.grade = grade;
19        this.GPA = GPA;
20    }
21 }
```

```
<terminated> Student [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 5:09:41 pm – 5:10:18 pm) [pid: 14592]
Enter name: Vishwas
Enter age: 22
Enter gender: M
Enter grade: 10
Enter GPA: 10
Name: Vishwas
Age: 22
Gender: M
Grade: 10
GPA: 10.0
```

3. Create a class called "Circle" that has the following properties: radius, diameter, and area. Include a constructor and methods to calculate the diameter and area of the circle.

<https://codeshare.io/eV6xll>

The screenshot shows an IDE interface with two tabs: 'Circle.java' and 'Console'. The code in 'Circle.java' defines a class 'Circle' with private fields for radius, diameter, and area, and methods to calculate the diameter and area. The 'Console' tab shows the output of running the program, where it prompts for the radius of the circle and then prints out the diameter and area.

```
1 package assignment2;
2 import java.util.Scanner;
3
4 public class Circle {
5
6     double radius;
7     double diameter;
8     double area;
9
10
11    public Circle(double radius) {
12        this.radius = radius;
13    }
14
15
16    public void calculateDiameter() {
17        diameter = radius * 2;
18        System.out.println("Diameter of the circle is: " + diameter);
19    }
20
21
22    public void calculateArea() {
23        area = Math.PI * radius * radius;
24        System.out.println("Area of the circle is: " + area);
25    }
26}
```

```
<terminated> Circle [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 12:39:19 pm – 12:39:23 pm) [pid: 8152]
Enter the radius of the circle: 10
Diameter of the circle is: 20.0
Area of the circle is: 314.1592653589793
```

4. Create a class called "Rectangle" that has the following properties: length, width, and area. Include a constructor and a method to calculate the area of the rectangle.

<https://codeshare.io/oQ3xRb>

The screenshot shows an IDE interface with two tabs: 'Rectangle.java' and 'Console'. The 'Rectangle.java' tab contains Java code for a Rectangle class. The 'Console' tab shows the application's output after running it with input values of 10 and 20, resulting in an output of 200.0.

```
36     public double getWidth() {
37         return this.width;
38     }
39
40     public double getArea() {
41         return this.area;
42     }
43
44     public static void main(String[] args) {
45         Scanner scanner = new Scanner(System.in);
46
47         System.out.print("Enter length: ");
48         double length = scanner.nextDouble();
49
50         System.out.print("Enter width: ");
51         double width = scanner.nextDouble();
52
53         Rectangle rectangle = new Rectangle(length, width);
54         rectangle.calculateArea();
55
56         System.out.println("Area of the rectangle is: " + rectangle.getArea());
57     }
58 }
59
60
61
```

```
Console × Problems Debug Shell
<terminated> Rectangle [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 1:28:40 pm - 1:28:46 pm) [pid: 20644]
Enter length: 10
Enter width: 20
Area of the rectangle is: 200.0
```

5. Create a class called "BankAccount" that has the following properties: account number, account balance, account holder name, and account type. Include a constructor and methods to deposit and withdraw money from the account.

<https://codeshare.io/yo0Yen>

The screenshot shows an IDE interface with two tabs: 'Rectangle.java' and 'BankAccount.java'. The 'BankAccount.java' tab contains Java code for a BankAccount class. The 'Console' tab shows the application's output after running it with various inputs for account details and transaction amounts, resulting in successful deposit and withdrawal messages.

```
36         int accountNumber = scanner.nextInt();
37
38         System.out.println("Enter Account Balance: ");
39         double accountBalance = scanner.nextDouble();
40
41         System.out.println("Enter Account Holder Name: ");
42         String accountHolderName = scanner.next();
43
44         System.out.println("Enter Account Type: ");
45         String accountType = scanner.next();
46
47         BankAccount account = new BankAccount(accountNumber, accountBalance, accountHolderName, acco
48
49         System.out.println("Enter amount to deposit: ");
50         double amountToDeposit = scanner.nextDouble();
51         account.deposit(amountToDeposit);
```

```
Console × Problems Debug Shell
<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 1:32:08 pm - 1:32:40 pm) [pid: 10544]
Enter Account Number:
122
Enter Account Balance:
0
Enter Account Holder Name:
Vishwas
Enter Account Type:
Saving
Enter amount to deposit:
1000
1000.0 deposited successfully. Your new account balance is 1000.0
Enter amount to withdraw:
500
500.0 withdrawn successfully. Your new account balance is 500.0
```

6. Create a class called "Person" that has the following properties: name, age, address, phone number, and email address. Include a constructor and getter and setter methods for each property.

<https://codeshare.io/Ad1Jq8>

The screenshot shows an IDE interface with two tabs: "Person.java" and "Console".

Person.java:

```
64     String name = input.nextLine();
65
66     System.out.print("Enter age: ");
67     int age = input.nextInt();
68     input.nextLine();
69
70     System.out.print("Enter address: ");
71     String address = input.nextLine();
72
73     System.out.print("Enter phone number: ");
74     String phoneNumber = input.nextLine();
75
76     System.out.print("Enter email address: ");
77     String emailAddress = input.nextLine();
78
79     Person person = new Person(name, age, address, phoneNumber, emailAddress);
80
81     System.out.println("\nName: " + person.getName());
82     System.out.println("Age: " + person.getAge());
83     System.out.println("Address: " + person.getAddress());
```

Console:

```
<terminated> Person [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 2:31:00 pm - 2:31:36 pm) [pid: 760]
Enter name: Vishwas
Enter age: 22
Enter address: Mandya
Enter phone number: 7204936863
Enter email address: vishwas.chandra@tecnottree.com

Name: Vishwas
Age: 22
Address: Mandya
Phone number: 7204936863
Email address: vishwas.chandra@tecnottree.com
```

7. Create a class called "Animal" that has the following properties: name, species, age, and weight. Include a constructor and getter and setter methods for each property.

<https://codeshare.io/gL9xZy>

The screenshot shows an IDE interface with two tabs: "Animal.java" and "Console".

Animal.java:

```
52
53     // Main method to take user input and create Animal object
54●  public static void main(String[] args) {
55     Scanner input = new Scanner(System.in);
56
57     System.out.print("Enter name: ");
58     String name = input.nextLine();
59
60     System.out.print("Enter species: ");
61     String species = input.nextLine();
62
63     System.out.print("Enter age: ");
64     int age = input.nextInt();
65
66     System.out.print("Enter weight: ");
67     double weight = input.nextDouble();
68
69     // Create Animal object using user input
70     Animal animal = new Animal(name, species, age, weight);
71
72     // Print out animal object properties
73     System.out.println("Name: " + animal.getName());
74     System.out.println("Species: " + animal.getSpecies());
```

Console:

```
<terminated> Animal [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 2:40:58 pm - 2:41:38 pm) [pid: 9948]
Enter name: Cat
Enter species: Mammals
Enter age: 10
Enter weight: 10
Name: Cat
Species: Mammals
Age: 10
Weight: 10.0
```

8. Create a class called "Triangle" that has the following properties: base, height, and area. Include a constructor and a method to calculate the area of the triangle.

<https://codeshare.io/xv4Y9R>

The screenshot shows a Java IDE interface with two tabs: "Triangle.java" and "Console".

Triangle.java:

```
38     this.height = height;
39     this.area = calculateArea();
40 }
41
42 public double getArea() {
43     return area;
44 }
45
46 // Main method to take user input and create Triangle object
47 public static void main(String[] args) {
48     Scanner input = new Scanner(System.in);
49
50     System.out.print("Enter base length: ");
51     double base = input.nextDouble();
52
53     System.out.print("Enter height length: ");
54     double height = input.nextDouble();
55
56     // Create Triangle object using user input
57     Triangle triangle = new Triangle(base, height);
58
59     // Print out triangle object properties
60     System.out.println("Base: " + triangle.getBase());
61     System.out.println("Height: " + triangle.getHeight());
62     System.out.println("Area: " + triangle.getArea());
63 }
64 }
```

Console:

```
<terminated> Triangle [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 3:20:18 pm – 3:20:25 pm) [pid: 12128]
Enter base length: 10
Enter height length: 20
Base: 10.0
Height: 20.0
Area: 100.0
```

9. Create a class called "Employee" that has the following properties: name, employee ID, department, job title, and salary. Include a constructor and getter and setter methods for each property.

<https://codeshare.io/pqkxp4>

The screenshot shows a Java IDE interface with two tabs: "Employee.java" and "Console".

Employee.java:

```
65     scanner.nextLine();
66
67     System.out.print("Enter employee department: ");
68     String department = scanner.nextLine();
69
70     System.out.print("Enter employee job title: ");
71     String jobTitle = scanner.nextLine();
72
73     System.out.print("Enter employee salary: ");
74     double salary = scanner.nextDouble();
75
76     Employee employee = new Employee(name, employeeID, department, jobTitle, salary);
77
78     System.out.println("Employee name: " + employee.getName());
79     System.out.println("Employee ID: " + employee.getEmployeeID());
80     System.out.println("Employee department: " + employee.getDepartment());
81     System.out.println("Employee job title: " + employee.getJobTitle());
82     System.out.println("Employee salary: " + employee.getSalary());
83
84 }
```

Console:

```
<terminated> Employee [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 3:23:52 pm – 3:24:38 pm) [pid: 14888]
Enter employee name: Vishwas
Enter employee ID: 20694
Enter employee department: Value Engineering
Enter employee job title: Engineering Intern
Enter employee salary: 38000
Employee name: Vishwas
Employee ID: 20694
Employee department: Value Engineering
Employee job title: Engineering Intern
Employee salary: 38000.0
```

10. Create a class called "Address" that has the following properties: street, city, state, zip code, and country. Include a constructor and getter and setter methods for each property.

<https://codeshare.io/BA7XEx>

The screenshot shows a Java development environment with two open files: `Address.java` and `ArrayListExample.java`. The `Address.java` file contains code to read user input for street, city, state, ZIP code, and country, and then prints them out. The `ArrayListExample.java` file is partially visible. Below the editor is a `Console` window showing the execution of the `Address` application. The console output is as follows:

```
<terminated> Address [Java Application] C:\Program Files\Java\jdk-19\bin\javaw.exe (28-Feb-2023, 3:34:03 pm – 3:36:13 pm) [pid: 20588]
Enter street address: Arakeshwara Nagar
Enter city: Mandyā
Enter state: Karnataka
Enter ZIP code: 571403
Enter country: India
Street address: Arakeshwara Nagar
City: Mandyā
State: Karnataka
ZIP code: 571403
Country: India
```