

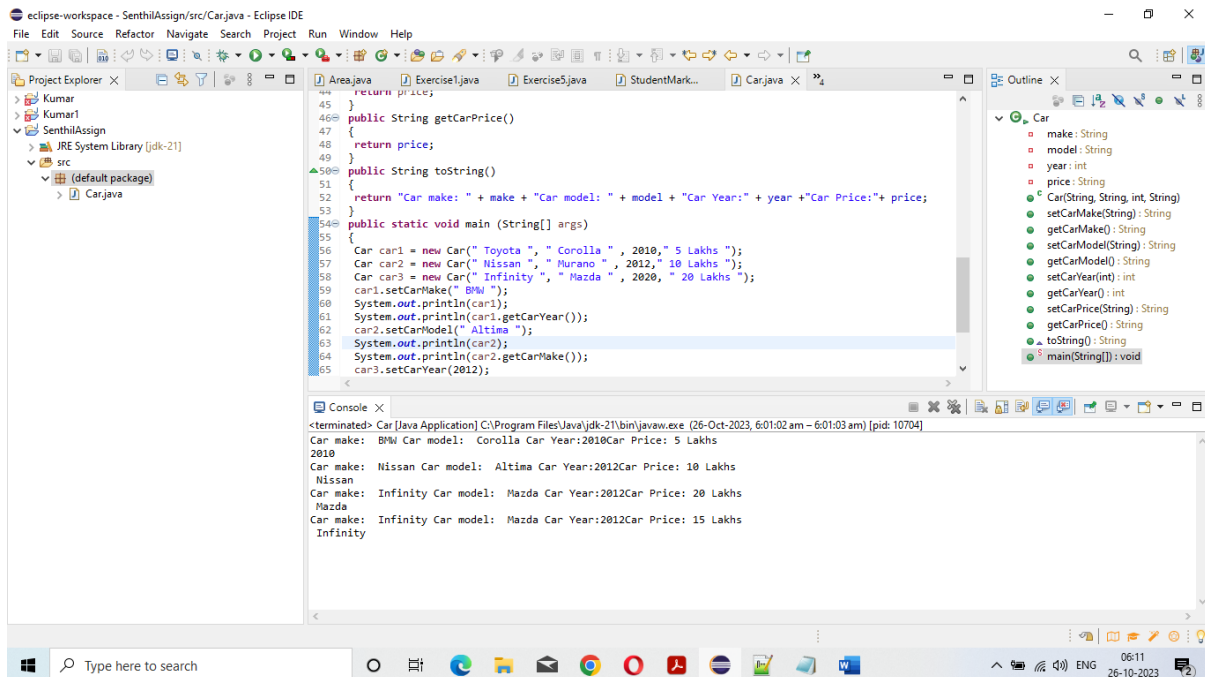
Assignment 2

[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.

Codeshare Link

<https://codeshare.io/mpApYj>

Screenshot



The screenshot shows the Eclipse IDE interface with the following components:

- Project Explorer:** Shows a project named 'SenthilAssign' with a source folder 'src' containing a file 'Car.java'.
- Editor:** Displays the code for 'Car.java'. The code includes a constructor, getters, setters, and a main method that creates three car objects and prints their details.
- Outline:** Lists the methods and fields of the 'Car' class.
- Console:** Shows the output of the program, displaying the details of three cars: a BMW Corolla, a Nissan Altima, and an Infinity Mazda.

```
44     return price;
45 }
46 public String getCarPrice()
47 {
48     return price;
49 }
50 public String toString()
51 {
52     return "Car make: " + make + "Car model: " + model + "Car Year:" + year + "Car Price:" + price;
53 }
54 public static void main (String[] args)
55 {
56     Car car1 = new Car(" Toyota ", " Corolla ", 2010, " 5 Lakhs ");
57     Car car2 = new Car(" Nissan ", " Murano ", 2012, " 10 Lakhs ");
58     Car car3 = new Car(" Infinity ", " Mazda ", 2020, " 20 Lakhs ");
59     car1.setCarMake(" BMW ");
60     System.out.println(car1);
61     System.out.println(car1.getCarYear());
62     car2.setCarModel(" Altima ");
63     System.out.println(car2);
64     System.out.println(car2.getCarMake());
65     car3.setCarYear(2012);
```

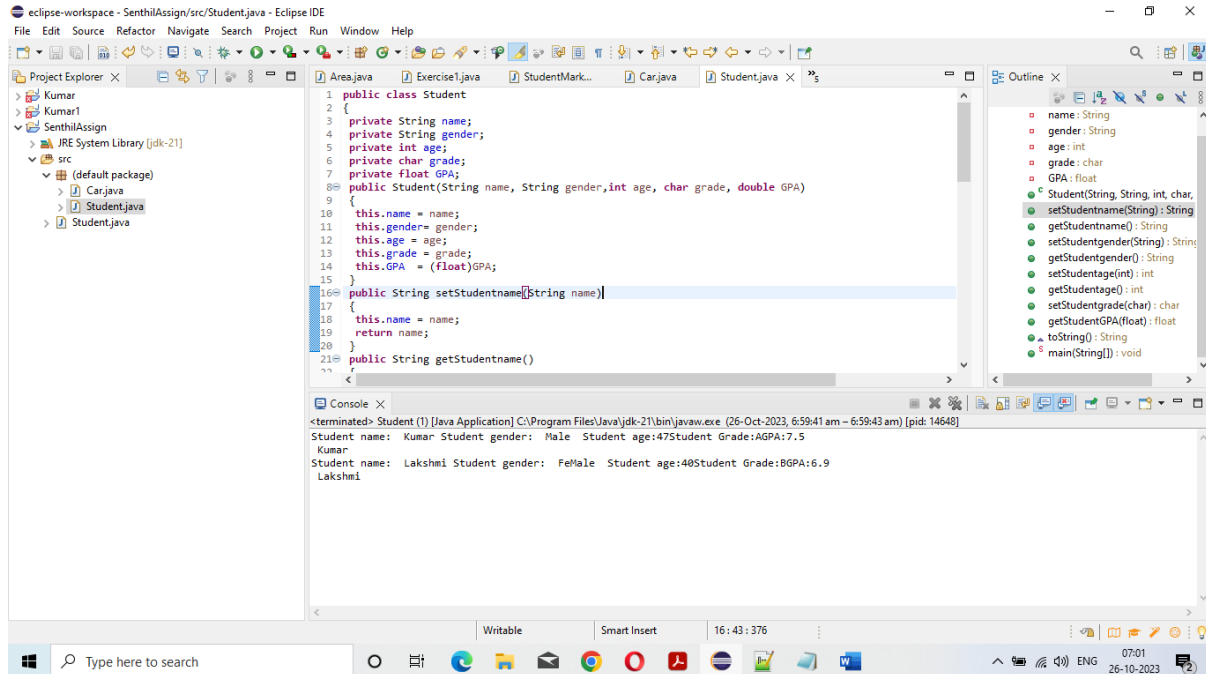
Console Output:

```
<terminated> Car [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (26-Oct-2023, 6:01:02 am - 6:01:03 am) [pid: 10704]
Car make: BMW Car model: Corolla Car Year:2010Car Price: 5 Lakhs
2010
Car make: Nissan Car model: Altima Car Year:2012Car Price: 10 Lakhs
Nissan
Car make: Infinity Car model: Mazda Car Year:2012Car Price: 20 Lakhs
Mazda
Car make: Infinity Car model: Mazda Car Year:2012Car Price: 15 Lakhs
Infinity
```

[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.

Codeshare link

<https://codeshare.io/JbZb46>



The screenshot shows the Eclipse IDE interface. The main editor displays the code for the `Student` class in `Student.java`. The code includes private fields for `name`, `gender`, `age`, `grade`, and `GPA`, a constructor, and getter/setter methods for each field. The `setStudentname` method is currently selected. The Outline view on the right lists all the methods and fields. The Console view at the bottom shows the output of the program, which prints the details for two students: Kumar and Lakshmi.

```
1 public class Student
2 {
3     private String name;
4     private String gender;
5     private int age;
6     private char grade;
7     private float GPA;
8     public Student(String name, String gender, int age, char grade, double GPA)
9     {
10         this.name = name;
11         this.gender = gender;
12         this.age = age;
13         this.grade = grade;
14         this.GPA = (float)GPA;
15     }
16     public String setStudentname(String name)
17     {
18         this.name = name;
19         return name;
20     }
21     public String getStudentname()
22     {
23         return name;
24     }
25 }
```

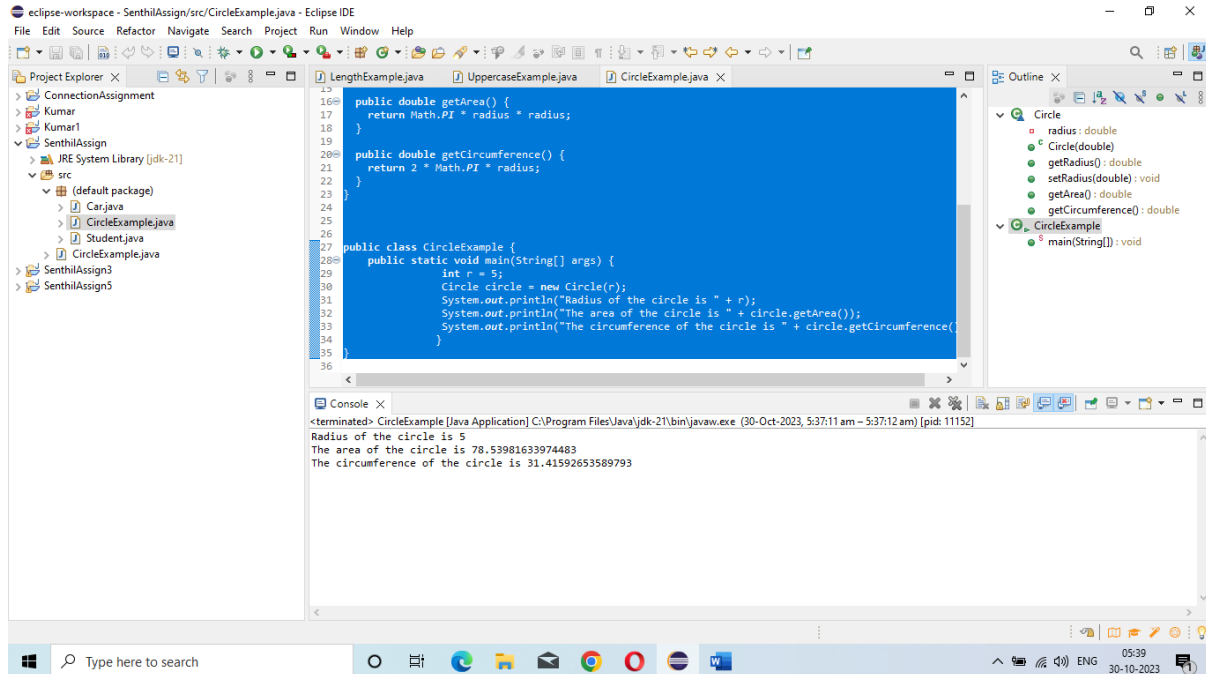
Console Output:

```
<terminated> Student (1) [Java Application] C:\Program Files\Java\jdk-21\bin\java.exe (26-Oct-2023, 6:59:41 am - 6:59:43 am) [pid: 14648]
Student name: Kumar Student gender: Male Student age:47Student Grade:AGPA:7.5
Kumar
Student name: Lakshmi Student gender: Female Student age:40Student Grade:BGPA:6.9
Lakshmi
```

[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

Codeshare link:

<https://codeshare.io/dwmpZB>



The screenshot shows the Eclipse IDE interface. The main editor displays the `CircleExample.java` file with the following code:

```
16 public double getArea() {
17     return Math.PI * radius * radius;
18 }
19
20 public double getCircumference() {
21     return 2 * Math.PI * radius;
22 }
23
24
25
26
27 public class CircleExample {
28     public static void main(String[] args) {
29         int r = 5;
30         Circle circle = new Circle(r);
31         System.out.println("Radius of the circle is " + r);
32         System.out.println("The area of the circle is " + circle.getArea());
33         System.out.println("The circumference of the circle is " + circle.getCircumference());
34     }
35 }
36
```

The right-hand side of the IDE shows the Outline view with the following structure:

- Circle
 - radius : double
 - Circle(double)
 - getRadius() : double
 - setRadius(double) : void
 - getArea() : double
 - getCircumference() : double
- CircleExample
 - main(String[]) : void

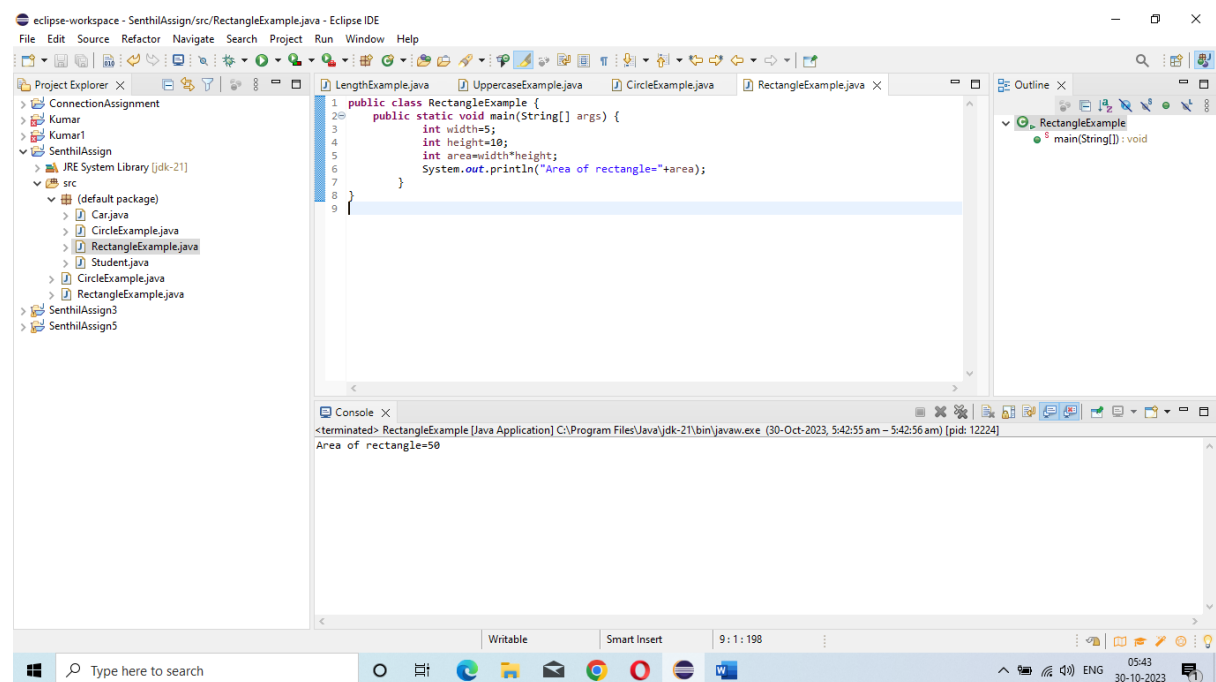
The bottom console window shows the output of the program:

```
<terminated> CircleExample [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (30-Oct-2023, 5:37:11 am - 5:37:12 am) [pid: 11152]
Radius of the circle is 5
The area of the circle is 78.53981633974483
The circumference of the circle is 31.41592653589793
```

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.

Codeshare Link:

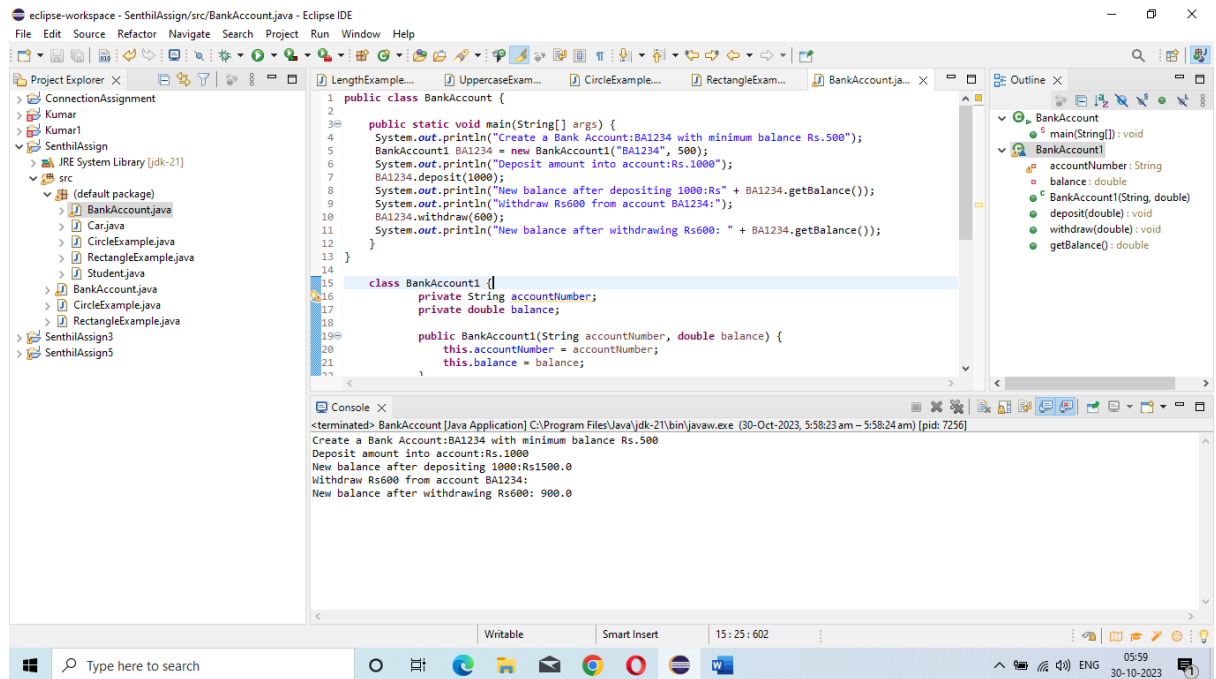
<https://codeshare.io/9OoYW7>



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.

Codeshare link:

<https://codeshare.io/3Aex41>



The screenshot displays the Eclipse IDE interface. The main editor window shows the `BankAccount.java` file with the following code:

```
1 public class BankAccount {
2
3     public static void main(String[] args) {
4         System.out.println("Create a Bank Account:BA1234 with minimum balance Rs.500");
5         BankAccount1 BA1234 = new BankAccount1("BA1234", 500);
6         System.out.println("Deposit amount into account:Rs.1000");
7         BA1234.deposit(1000);
8         System.out.println("New balance after depositing 1000:Rs" + BA1234.getBalance());
9         System.out.println("Withdraw Rs600 from account BA1234:");
10        BA1234.withdraw(600);
11        System.out.println("New balance after withdrawing Rs600: " + BA1234.getBalance());
12    }
13 }
14
15 class BankAccount1 {
16     private String accountNumber;
17     private double balance;
18
19     public BankAccount1(String accountNumber, double balance) {
20         this.accountNumber = accountNumber;
21         this.balance = balance;
22     }
23 }
```

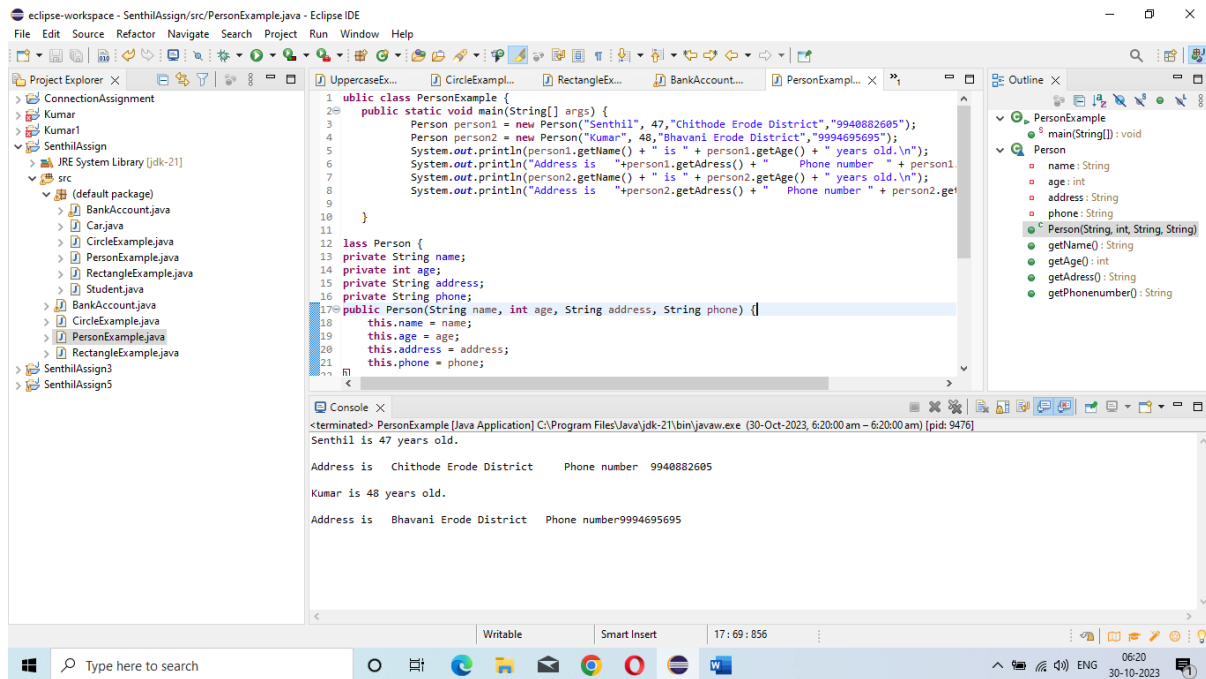
The Project Explorer on the left shows the project structure, including the `BankAccount.java` file. The Console window at the bottom displays the output of the program:

```
<terminated> BankAccount [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (30-Oct-2023, 5:58:23 am - 5:58:24 am) [pid: 7256]
Create a Bank Account:BA1234 with minimum balance Rs.500
Deposit amount into account:Rs.1000
New balance after depositing 1000:Rs1500.0
Withdraw Rs600 from account BA1234:
New balance after withdrawing Rs600: 900.0
```

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.

Codeshare link:

<https://codeshare.io/MNqJLR>



The screenshot shows the Eclipse IDE interface. The main editor displays the `PersonExample.java` file, which contains the following code:

```
1 public class PersonExample {
2     public static void main(String[] args) {
3         Person person1 = new Person("Senthil", 47, "Chithode Erode District", "9948882605");
4         Person person2 = new Person("Kumar", 48, "Bhavani Erode District", "9994695695");
5         System.out.println(person1.getName() + " is " + person1.getAge() + " years old.\n");
6         System.out.println("Address is " + person1.getAddress() + " Phone number " + person1
7         System.out.println(person2.getName() + " is " + person2.getAge() + " years old.\n");
8         System.out.println("Address is " + person2.getAddress() + " Phone number " + person2.get
9     }
10 }
11
12 class Person {
13     private String name;
14     private int age;
15     private String address;
16     private String phone;
17     public Person(String name, int age, String address, String phone) {
18         this.name = name;
19         this.age = age;
20         this.address = address;
21         this.phone = phone;
22     }
23 }
```

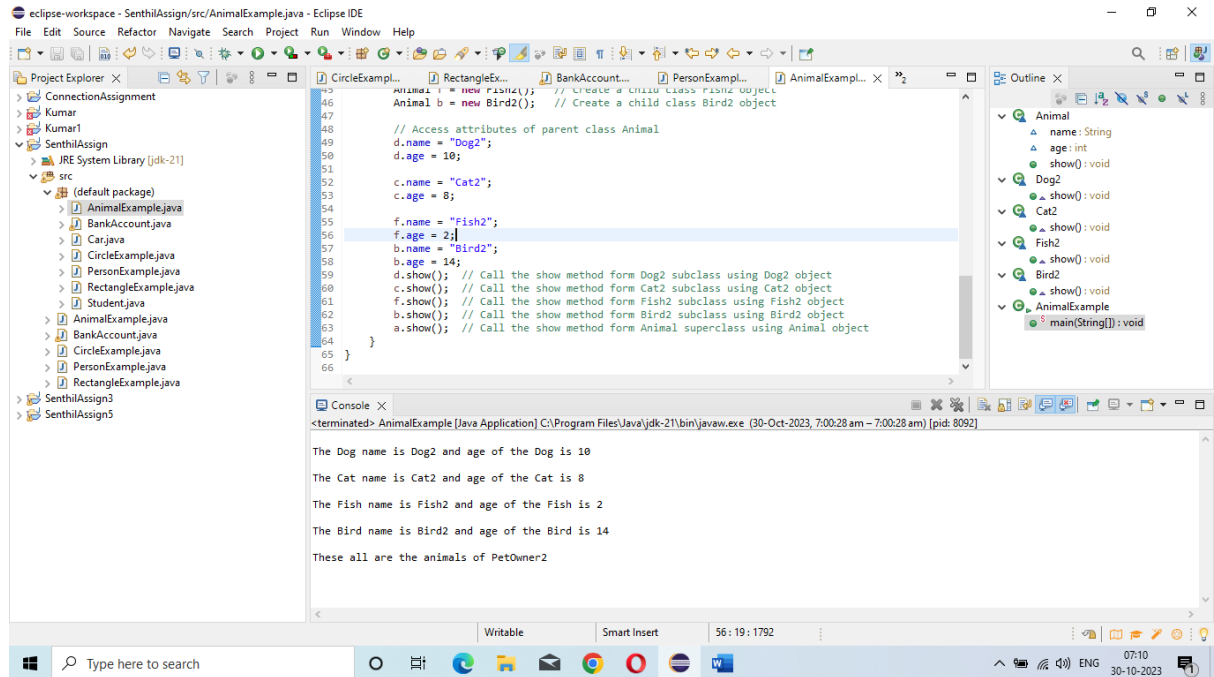
The console output shows the execution results:

```
<terminated> PersonExample [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (30-Oct-2023, 6:20:00 am - 6:20:00 am) [pid: 9476]
Senthil is 47 years old.
Address is Chithode Erode District Phone number 9948882605
Kumar is 48 years old.
Address is Bhavani Erode District Phone number 9994695695
```

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.

Codeshare link:

<https://codeshare.io/PdWz9M>



The screenshot displays the Eclipse IDE interface. The central editor shows the `AnimalExample.java` file with the following code:

```
46 Animal a = new Fish2(); // Create a child class Fish2 object
47 Animal b = new Bird2(); // Create a child class Bird2 object
48
49 // Access attributes of parent class Animal
50 d.name = "Dog2";
51 d.age = 10;
52
53 c.name = "Cat2";
54 c.age = 8;
55
56 f.name = "Fish2";
57 f.age = 2;
58 b.name = "Bird2";
59 b.age = 14;
60 d.show(); // Call the show method form Dog2 subclass using Dog2 object
61 c.show(); // Call the show method form Cat2 subclass using Cat2 object
62 f.show(); // Call the show method form Fish2 subclass using Fish2 object
63 b.show(); // Call the show method form Bird2 subclass using Bird2 object
64 a.show(); // Call the show method form Animal superclass using Animal object
65 }
66 }
```

The right-hand side of the IDE shows the Outline view with the following class hierarchy:

- Animal
 - name: String
 - age: int
 - show(): void
- Dog2
 - show(): void
- Cat2
 - show(): void
- Fish2
 - show(): void
- Bird2
 - show(): void
- AnimalExample
 - main(String[]): void

The bottom console window shows the output of the program:

```
<terminated> AnimalExample [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (30-Oct-2023, 7:00:28 am - 7:00:28 am) [pid: 8092]

The Dog name is Dog2 and age of the Dog is 10

The Cat name is Cat2 and age of the Cat is 8

The Fish name is Fish2 and age of the Fish is 2

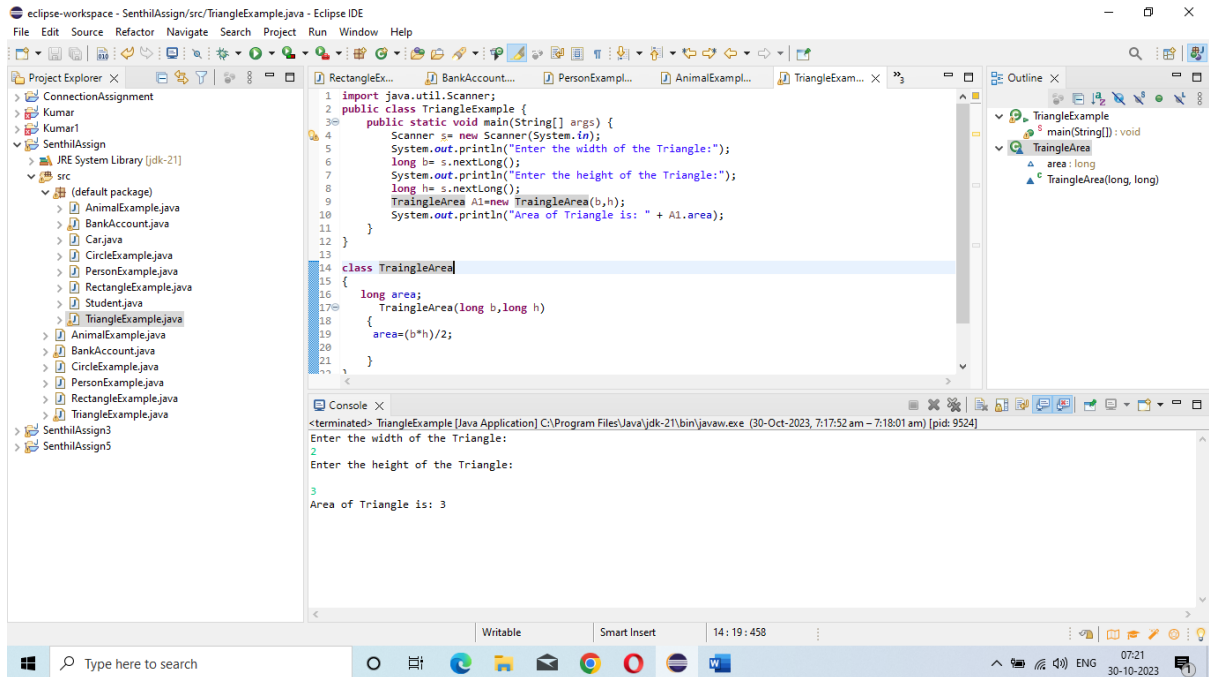
The Bird name is Bird2 and age of the Bird is 14

These all are the animals of PetOwner2
```

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.

Codeshare link:

<https://codeshare.io/6p8dOY>



The screenshot displays the Eclipse IDE interface. The main editor window shows the `TriangleExample.java` file with the following code:

```
1 import java.util.Scanner;
2 public class TriangleExample {
3     public static void main(String[] args) {
4         Scanner s = new Scanner(System.in);
5         System.out.println("Enter the width of the Triangle:");
6         long b = s.nextLong();
7         System.out.println("Enter the height of the Triangle:");
8         long h = s.nextLong();
9         TriangleArea A1 = new TriangleArea(b,h);
10        System.out.println("Area of Triangle is: " + A1.area);
11    }
12 }
13
14 class TriangleArea
15 {
16     long area;
17     TriangleArea(long b,long h)
18     {
19         area=(b*h)/2;
20     }
21 }
```

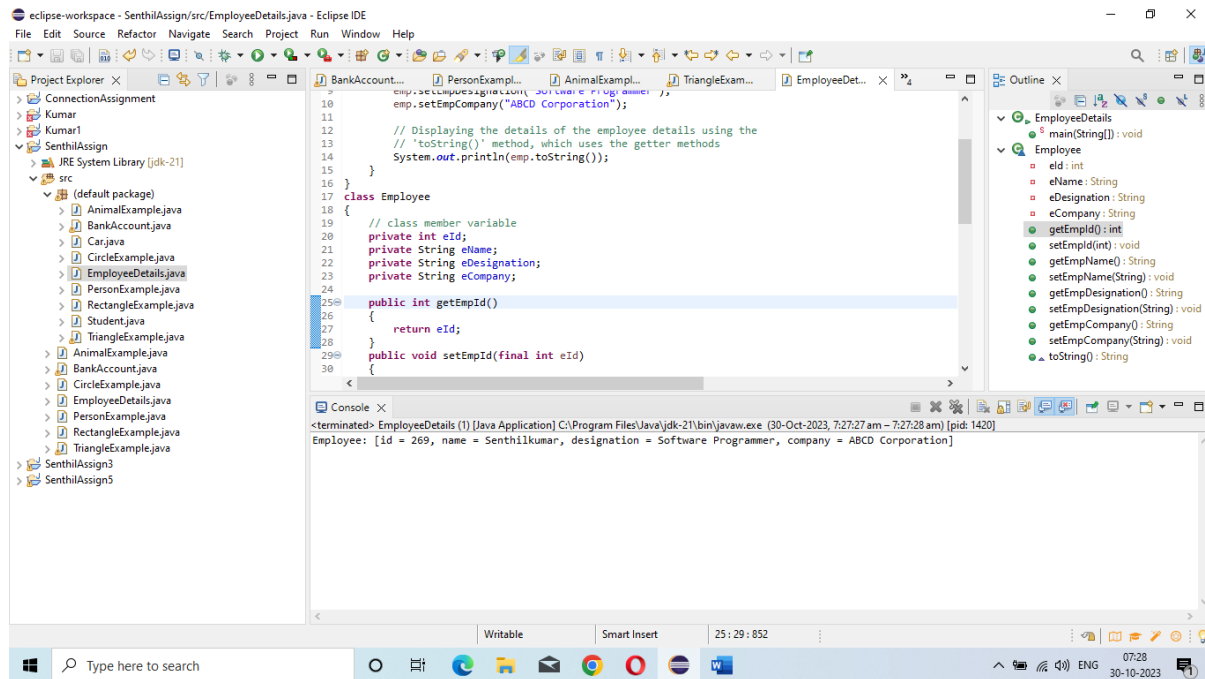
The Project Explorer on the left shows the project structure, including the `TriangleExample.java` file. The Console window at the bottom shows the execution output:

```
<terminated> TriangleExample [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe (30-Oct-2023, 7:17:52 am - 7:18:01 am) [pid: 9524]
Enter the width of the Triangle:
2
Enter the height of the Triangle:
3
Area of Triangle is: 3
```


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.

Codeshare link:

<https://codeshare.io/nz1X0Y>



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.

Codeshare link:

<https://codeshare.io/eVlJB4>

The screenshot displays the Eclipse IDE interface with the following components:

- Project Explorer:** Shows a project named 'SenthilAssign' with a source folder 'src' containing several Java files, including 'AddressExample.java'.
- Editor:** Displays the code for 'AddressExample.java'. The code defines an 'Address' class with attributes: 'doornumber' (int), 'name' (String), 'streetname' (String), 'cityname' (String), 'statename' (String), 'pincode' (String), and 'countryname' (String). It includes getter and setter methods for each attribute and a 'toString()' method. The main method in 'AddressExample' creates an 'Address' object with specific values and prints it.
- Outline:** Shows the structure of the 'Address' class with its attributes and methods.
- Console:** Displays the output of the program execution, showing the details of the created 'Address' object: 'Address: [Doornumber= 47, name = Senthilkumar, Streetname = Mudhaliyar Street, Cityname= Chithode, statename=Chithode, countryname=Chithode, pincode= 560002]'.

