Tuesday, October 8, 2024 12:06 PM

Assignment on super and this

Assignment 1: Using this Keyword for Constructor Overloading

Problem:

Create a class Person with the following properties:

- String name
- int age

Create two constructors:

1. One that takes both name and age as parameters.
2. Another that only takes name and sets a default value for age (e.g., 25) by using this() to call the first constructor.

Write a display method to print the values of name and age.

Requirements:

- Use the this() keyword to call one constructor from the other.
- Create an object using both constructors and call the display method.

Example Output:

```
Name: John, Age: 25
Name: Alice, Age: 30
```

Assignment 2: Using super Keyword to Access Parent Class Method

Problem:

Create two classes:

1. Parent class with a method show() that prints “This is the Parent class.”
2. Child class that overrides the show() method to print “This is the Child class.”

In the Child class, call the parent class’s show() method using super.

Requirements:

- Create an object of the Child class and call its show() method.
- The method should first print the message from the parent class, then the message from the child class.

Example Output:

```
This is the Parent class.
This is the Child class.
```

Assignment 3: Using super and this for Accessing Variables

Problem:

Create three classes:

1. Parent class with a variable String var = "Parent's variable".
2. Child class with a variable String var = "Child's variable".
3. GrandChild class with a variable String var = "GrandChild's variable".

In the GrandChild class:

- Use this.var to print its own variable.
- Use super.var to print the variable from the Child class.
- Add a method in Child class to access the Parent class's variable using super.var, and call this method from the GrandChild class.

Requirements:

- Print the variables from all three classes using this and super.

Example Output:

```
GrandChild's variable
Child's variable
Parent's variable
```

Assignment 4: Using this to Differentiate Between Instance and Local Variables

Problem:

Create a class Rectangle with the following properties:

- int length
- int width

Write a constructor that accepts parameters with the same names (length and width) as the instance variables. Use the this keyword to differentiate between the instance variables and the parameters.

Create a method area() to calculate the area of the rectangle and a method display() to display the values of length, width, and the area.

Requirements:

- Use the this keyword to assign constructor parameters to instance variables.

Example Output:

```
Length: 5, Width: 10, Area: 50
```

Assignment 5: Using super() to Call Parent Class Constructor

Problem:

Create two classes:

1. Animal class with a constructor that takes a String name as a parameter and prints "Animal: [name]".
2. Dog class that inherits from Animal and has its own constructor that also takes a String name as a parameter. Use super() to call the parent class constructor from the child class constructor. Additionally, print "Dog: [name]" in the Dog class constructor.

Requirements:

- Create an object of the Dog class and observe the output.
- Use super() to call the parent class constructor.

Example Output:

```
Animal: Buddy
Dog: Buddy
```

Assignment 6: Chaining Constructors Using this() and super()

Problem:

Create three classes: Grandparent, Parent, and Child.

1. Grandparent has a constructor that prints "Grandparent constructor".
2. Parent has a constructor that calls the Grandparent constructor using super() and prints "Parent constructor".
3. Child has a constructor that calls the Parent constructor using super() and prints "Child constructor".

In the Child class:

- Chain constructors using this() and super() and observe the order of constructor calls.

Requirements:

- Create an object of the Child class and print the order of constructor calls.

Example Output:

```
Grandparent constructor
Parent constructor
Child constructor
```

These assignments will help you get hands-on experience with this and super in various scenarios like constructor chaining, method overriding, and variable shadowing.