Assignment 1: Student Class (Class & Object)

```
class Student {
    int id;
    String name;

    void displayDetails() {
        System.out.println("ID: " + id);
        System.out.println("Name: " + name);
    }

    public static void main(String[] args) {
        Student s = new Student();
        s.id = 1;
        s.name = "Jayasree";

        s.displayDetails();
    }
}
```

Assignment 2: Employee Salary (Default Constructor)

```
class Employee {
    int empId;
    String empName;
    double salary;

    Employee() {
        empId = 101;
        empName = "Rahul";
        salary = 30000;
    }
```

```
    void display() {
        System.out.println(empId);
        System.out.println(empName);
        System.out.println(salary);
    }

    public static void main(String[] args) {
        Employee e = new Employee();
        e.display();
    }
}
```

Assignment 3: Car Information (Parameterized Constructor)

```
class Car {
    String brand;
    String model;
    double price;

    Car(String b, String m, double p) {
        brand = b;
        model = m;
        price = p;
    }

    void display() {
        System.out.println(brand + " " + model + " " + price);
    }

    public static void main(String[] args) {
```

```
        Car c1 = new Car("Toyota", "Innova", 2000000);
        Car c2 = new Car("Hyundai", "i20", 900000);

        c1.display();
        c2.display();
    }
}
```

Assignment 4: Book Details (Constructor Overloading)

```
class Book {
    String title;
    String author;

    Book() {
        title = "Unknown";
        author = "Unknown";
    }

    Book(String t, String a) {
        title = t;
        author = a;
    }

    void display() {
        System.out.println(title + " - " + author);
    }

    public static void main(String[] args) {
        Book b1 = new Book();
        Book b2 = new Book("Java", "James Gosling");
```

```
        b1.display();
        b2.display();
    }
}
```

Assignment 5: Bank Account (Object Initialization)

```
class BankAccount {
    int accountNumber;
    String holderName;
    double balance;

    BankAccount(int acc, String name, double bal) {
        accountNumber = acc;
        holderName = name;
        balance = bal;
    }

    void showAccount() {
        System.out.println(accountNumber);
        System.out.println(holderName);
        System.out.println(balance);
    }

    public static void main(String[] args) {
        BankAccount b = new BankAccount(12345, "Jayasree", 50000);
        b.showAccount();
    }
}
```

Assignment 6: Product Management (this keyword)

```
class Product {
    int productId;
    String productName;
    double price;

    Product(int productId, String productName, double price) {
        this.productId = productId;
        this.productName = productName;
        this.price = price;
    }

    void display() {
        System.out.println(productId);
        System.out.println(productName);
        System.out.println(price);
    }

    public static void main(String[] args) {
        Product p = new Product(1, "Laptop", 60000);
        p.display();
    }
}
```

Assignment 7: User Login System (Constructor Overloading)

```
class User {
    String email;
    String password;

    User(String email) {
        this.email = email;
```

```
        this.password = "Not Set";
    }

    User(String email, String password) {
        this.email = email;
        this.password = password;
    }

    void display() {
        System.out.println(email);
        System.out.println(password);
    }

    public static void main(String[] args) {
        User u1 = new User("abc@gmail.com");
        User u2 = new User("xyz@gmail.com", "12345");

        u1.display();
        u2.display();
    }
}
```

Assignment 8: Mobile Store (Multiple Constructors)

```
class Mobile {
    String brand;
    int ram;
    int storage;
    double price;

    Mobile(String brand) {
```

```
        this.brand = brand;
    }

    Mobile(String brand, int ram, int storage, double price) {
        this.brand = brand;
        this.ram = ram;
        this.storage = storage;
        this.price = price;
    }

    void display() {
        System.out.println(brand + " " + ram + "GB " + storage + "GB " + price);
    }

    public static void main(String[] args) {
        Mobile m1 = new Mobile("Samsung");
        Mobile m2 = new Mobile("Apple", 8, 128, 80000);

        m1.display();
        m2.display();
    }
}
```

Assignment 9: Library Book System (Object Behavior)

```
class LibraryBook {
    int bookId;
    String title;
    String author;
    boolean available;
```

```
    LibraryBook(int id, String t, String a) {
        bookId = id;
        title = t;
        author = a;
        available = true;
    }

    boolean isAvailable() {
        return available;
    }

    public static void main(String[] args) {
        LibraryBook b = new LibraryBook(1, "Java", "James");
        System.out.println(b.isAvailable());
    }
}
```

Assignment 10: College Admission (Constructor Logic)

```
class StudentAdmission {
    String admissionType;

    StudentAdmission() {
        admissionType = "General Admission";
    }

    StudentAdmission(String merit) {
        admissionType = "Merit Admission";
    }

    void display() {
```

```
        System.out.println(admissionType);
    }

    public static void main(String[] args) {
        StudentAdmission s1 = new StudentAdmission();
        StudentAdmission s2 = new StudentAdmission("Merit");

        s1.display();
        s2.display();
    }
}
```

Assignment 11: Constructor Chaining (this())

```
class Person {
    String name;
    int age;

    Person(String name) {
        this.name = name;
    }

    Person(String name, int age) {
        this(name);
        this.age = age;
    }

    void display() {
        System.out.println(name + " " + age);
    }
```

```
    public static void main(String[] args) {
        Person p = new Person("Jayasree", 20);
        p.display();
    }
}
```

Assignment 12: E-Commerce Order (Business Logic in Constructor)

```
class Order {
    int orderId;
    String customerName;
    double amount;

    Order(int id, String name, double amt) {
        orderId = id;
        customerName = name;
        amount = amt + (amt * 0.18); // 18% tax
    }

    void display() {
        System.out.println(orderId);
        System.out.println(customerName);
        System.out.println(amount);
    }

    public static void main(String[] args) {
        Order o = new Order(1, "Jayasree", 1000);
        o.display();
    }
}
```