**Exercise 1:**

**Objective:** Practice using the **public** access specifier.

1. Create a class named **PublicClass**.
2. Declare a public variable called **publicVariable** of type **int** with an initial value of 42.
3. Implement a public method called **publicMethod** that prints "Public method called."
4. In the **MainClass**, create an instance of **PublicClass** and use the public variable and method.

**Exercise 2:**

**Objective:** Practice using the **protected** access specifier.

1. Create a class named **ProtectedClass**.
2. Declare a protected variable called **protectedVariable** of type **String** with an initial value of "Hello."
3. Implement a protected method called **protectedMethod** that prints "Protected method called."
4. Create a subclass named **DerivedClass** that extends **ProtectedClass**. In **DerivedClass**, use the protected variable and method.
5. In the **MainClass**, create an instance of **DerivedClass** and use its members.

**Exercise 3:**

**Objective:** Practice using the default (package-private) access specifier.

1. Create a class named **DefaultClass**.
2. Declare a package-private variable called **defaultVariable** of type **double** with an initial value of 3.14.
3. Implement a package-private method called **defaultMethod** that prints "Default method called."
4. In the **MainClass**, create an instance of **DefaultClass** and use the default variable and method.

**Exercise 4:**

**Objective:** Practice using the **private** access specifier.

1. Create a class named **PrivateClass**.
2. Declare a private variable called **privateVariable** of type **boolean** with an initial value of **true**.
3. Implement a private method called **privateMethod** that prints "Private method called."
4. In the **MainClass**, try to create an instance of **PrivateClass** and observe the compilation error.

**Exercise 5:**

**Objective:** Simulate a real-world scenario using access specifiers.

Consider a scenario where you have a **Person** class with private fields for **name** and **age**. Implement getter and setter methods for these private fields. Additionally, create a subclass **Employee** that extends **Person**. The **Employee** class should have an additional private field for **employeeId**. Implement getter and setter methods for the **employeeId** field.

1. Create a class named **Person** with private fields **name** (String) and **age** (int).
2. Implement getter and setter methods for the **name** and **age** fields in the **Person** class.
3. Create a subclass named **Employee** that extends **Person**.
4. Add a private field **employeeId** (String) to the **Employee** class.
5. Implement getter and setter methods for the **employeeId** field in the **Employee** class.
6. In the **MainClass**, create an instance of **Employee**, set values using the setter methods, and retrieve values using the getter methods.

**Exercise 6:**

**Objective:** Simulate a real-world scenario using the default (package-private) access specifier.

Consider a scenario where you have a package with multiple classes representing a simple library. Implement a **Library** class with a package-private method to check the availability of a book by its title. Create a class named **Book** with private fields for **title** and **author**. The **Library** class should have a array of available books. The package-private method in **Library** should search for a book by title and return true if available; otherwise, return false.

1. Create a package (directory) named **library** for your classes.
2. Inside the **library** package, create a class named **Library** with a package-private method **isBookAvailable(String title)** that takes a book title as a parameter and returns a boolean indicating whether the book is available.
3. Inside the **library** package, create a class named **Book** with private fields **title** (String) and **author** (String).
4. In the **Library** class, maintain a array of available books. Initialize the list in the constructor with a few books.
5. In the **MainClass** (outside the **library** package), create an instance of **Library**, and use the **isBookAvailable()** method to check the availability of a book by title.

**Exercise 8:**

**Objective:** Simulate a real-world scenario using the default (package-private) access specifier with a practical example.

Consider a scenario where you have a package representing a basic customer management system. Implement a **Customer** class with package-private fields for **customerId** (String) and **customerName** (String). Create a **CustomerManager** class with a package-private method to add a new customer and check if a customer exists by their ID. The **CustomerManager** class should use an array or another suitable data structure to store customer information.

1. Create a package (directory) named **customer** for your classes.
2. Inside the **customer** package, create a class named **Customer** with package-private fields **customerId** and **customerName**.
3. Inside the **customer** package, create a class named **CustomerManager** with a package-private method **addCustomer(String customerId, String customerName)** that adds a new customer and a method **isCustomerExists(String customerId)** that checks if a customer exists by their ID.
4. In the **CustomerManager** class, use an array or another suitable data structure to store customer information. Initialize the data structure in the constructor.
5. In the **MainClass** (outside the **customer** package), create an instance of **CustomerManager**, add a few customers using the **addCustomer()** method, and check if a customer exists using the **isCustomerExists()** method.

**Exercise 9:**

**Objective:** Create a multi-level class hierarchy for an online shopping system using all access specifiers (default, private, protected, and public).

Consider a scenario where you have a package representing an online shopping system. Implement the following classes:

1. **Product (Public):**
   * Create a public class named **Product** with private fields for **productId** (String), **productName** (String), and **price** (double).
   * Implement a parameterized constructor to initialize these fields.
   * Provide public getter methods for each field.
2. **Customer (Public):**
   * Create a public class named **Customer** with private fields for **customerId** (String) and **customerName** (String).
   * Implement a parameterized constructor to initialize these fields.
   * Provide public getter methods for each field.
3. **Order (Default):**
   * Create a class named **Order** in the same package as **Product** and **Customer**.
   * Include private fields for **orderId** (String), **product** (Product), **quantity** (int), and **customer** (Customer).
   * Implement a parameterized constructor to initialize these fields.
   * Provide default (package-private) getter methods for each field.
4. **ShoppingCart (Protected):**
   * Create a class named **ShoppingCart** in the same package as **Product**, **Customer**, and **Order**.
   * Include a protected field for an array of **Order** objects.
   * Implement a protected method **calculateTotalPrice()** that calculates and returns the total price of all orders in the shopping cart.
   * Provide a protected method **printShoppingCart()** that prints details of all orders in the shopping cart.
5. **MainClass (Public):**
   * Create a public class named **MainClass** outside the package.
   * In the **MainClass**, create instances of **Product**, **Customer**, and **ShoppingCart**.
   * Demonstrate the usage of all access specifiers by accessing fields and methods from different classes.

**Exercise 10:**

**Objective:** Create a scenario using access specifiers to model a university registration system.

Consider a scenario where you have a package representing a university registration system. Implement the following classes:

1. **Student (Public):**
   * Create a public class named **Student** with private fields for **studentId** (String) and **studentName** (String).
   * Implement a parameterized constructor to initialize these fields.
   * Provide public getter methods for each field.
2. **Course (Default):**
   * Create a class named **Course** in the same package as **Student**.
   * Include private fields for **courseId** (String), **courseName** (String), and **creditHours** (int).
   * Implement a parameterized constructor to initialize these fields.
   * Provide default (package-private) getter methods for each field.
3. **Registration (Protected):**
   * Create a class named **Registration** in the same package as **Student** and **Course**.
   * Include a protected field for an array of **Course** objects.
   * Implement a protected method **addCourse(Course course)** that adds a course to the registration list.
   * Provide a protected method **printRegistration()** that prints details of all courses in the registration.
4. **University (Public):**
   * Create a public class named **University** in the same package as **Student**, **Course**, and **Registration**.
   * Include private fields for an array of **Student** objects and an array of **Registration** objects.
   * Implement a parameterized constructor to initialize these fields.
   * Provide public methods to add a student, add a course to a student's registration, and print the registration details of a student.
5. **MainClass (Public):**
   * Create a public class named **MainClass** outside the package.
   * In the **MainClass**, create instances of **University**, **Student**, **Course**, and **Registration**.
   * Demonstrate the usage of all access specifiers by accessing fields and methods from different classes.