CSC 220 [03] – Data Structures

Mini Project 01

Contribution Statement

Team 2

Connor Trimble, Gavin Hardison, Eunhui Cho

<Individual Contributions>

Connor Trimble - Person Class

- Declared four private instance variables: firstName, lastName, age, and address
- Implemented private static variable personCount to track total Person objects created
- Created constructor that accepts four parameters (firstName, lastName, age, address) and initializes all instance variables
- Implemented constructor to increment personCount each time a Person object is created
- Developed getDetails() method that prints person's basic information (first name, last name, age, address)
- Implemented getter methods for all instance variables: getFirstName(), getLastName(), getAge(), getAddress()
- Implemented setter methods for all instance variables: setFirstName(), setLastName(), setAge(), setAddress()
- Created static getPersonCount() method that returns the current value of personCount

Gavin Hardison - Student Class

- Extended Person class using inheritance
- Declared two private instance variables: studentId and major
- Implemented constructor with six parameters (firstName, lastName, age, address, studentId, major)
- Used super keyword to invoke Person class constructor with inherited attributes
- Initialized student-specific variables (studentId, major) in constructor
- Overrode getDetails() method with @Override annotation
- Called super.getDetails() to display person information, then added student ID and major
- Implemented method overloading with getDetails(boolean includeMajor)
- Created conditional logic in overloaded method to display major only when includeMajor is true
- Implemented getter and setter methods for studentId and major variables

Eunhui Cho - Main Class

- Created two Person objects (person1 and person2) with sample data
- Created one Student object (student1) with sample data including student ID and major
- Called getDetails() on person1 to display Person information
- Called getDetails() on person2 to display second Person's information
- Demonstrated method overriding by calling getDetails() on student1 object
- Demonstrated method overloading by calling getDetails(false) on student1 object
- Displayed total number of Person objects created using Person.getPersonCount()

AI Tool Usage:

No AI tools were used for code implementation.

Used for documentation assistance:

Prompts used : "How can I describe Person / Student / Main class implementation?”

Applied to : Formatting the structure of the ContributionStatement