

CPS2231: Computer Programming

Homework 5

Due Date: **Listed on Canvas**

1. Theoretical Questions:

- a) Explain what Inheritance is in your own words.
- b) What keyword do you use to define a subclass?
- c) Explain Method Overriding in your own words. What is its difference with Method Overloading?
- d) What is the `@Override` annotation for?
- e) What is multiple inheritance? Does Java support multiple inheritance from class?
- f) What is Polymorphism?
- g) What is the difference between Dynamic Binding and Static Binding? Provide an example of each.
- h) What is the protected modifier?
- i) What is the final modifier?

2. Coding exercise with Inheritance:

Part 1. Create a Parent Class

Using the SAMPLE CODE PROVIDED (Computer.java), create a complete Java class that can be used to create a **Computer** object as described below. DO NOT MODIFY THE EXISTING CODE in Computer.java; just add to it.

A **Computer** has a:

- Manufacturer
- Disk Size
- Manufacturing Date
- Number of cores

- a) Add all instance variables (data encapsulation is expected).
- b) The class must have getters and setters for all instance variables.
- c) The class must have two constructors: a no-arg constructor and a constructor that receives input parameters for each instance variable.
- d) Implement a toString() method for your class. (already implemented; DO NOT MODIFY)

Part 2. Create two Child Classes

Using the SAMPLE CODE PROVIDED (Desktop.java and Laptop.java), create two Java classes, **Desktop** and **Laptop** that both are **subclasses** of Computer.

- a) A **Desktop** has additional instance variables: **width** and **height** (data encapsulation is expected)
- b) A **Laptop** has an additional instance variable, **weight** (data encapsulation is expected). c) Write constructors for both classes that require input for all their instance variables (including **Manufacturing Date**, **Disk size**, **Number of cores** and **Color**) and no-args constructors.
- d) Create getters and setters for all instance variables for both Child classes.

e) Create toString() methods in Desktop and Laptop. (already implemented;
DO NOT MODIFY)

Part 3. Test Your Code in Gradescope

Test the parent class and the two subclasses you created (Computer.java, Desktop.java, and Laptop.java) using the Gradescope utility in Canvas. Submit ONLY the following files to

Assignment 5: Inheritance and Polymorphism in Gradescope in Canvas:

- Computer.java
- Desktop.java
- Laptop.java

Review the output of the test cases run on your code.

3. UML diagram

Create a UML class hierarchy (three connected UML diagrams) that has all three classes you created above. Make sure each UML diagram is one-to-one to the code, it matches the Java code exactly. **Make sure that both Child Classes point to their Parent.**

Hint: *your subclasses are siblings.*

Hint: *You need to create three UML diagrams for three created Java classes and connect them using arrows, where each Child Class points to its Parent. You can either draw this by hand and take a picture or use a tool /software of your choice. Here is a free online UML tool you could use: https://www.lucidchart.com/pages/examples/uml_diagram_tool*

Submit the following things in a single PDF file to Canvas:

- (1) Your answers to theoretical questions.
- (2) Your UML diagram.

You can save to PDF from Word or use Google doc and download as PDF.

Submit your source code to Canvas. There should be **four** .java files in total.

Note: **properly document all your programs.** Each of your programs should start with the following comment lines:

```
/******  
** *  
* Kean University  
* Spring 2024  
* Course: CPS*2231 - Computer Organization & Programming *  
* Author: Your Full Name, course, section *  
* Homework: 5, task 1  
* Program Description:  
*****/
```