48024
Applications Programming Assignment 1

Topics:

OO Design, Standard Patterns, Lists

Learning Outcomes:

This assessment task addresses the following subject learning objectives (SLOs): 1, 2 and 3

Due date:

11:59PM Monday the 16th of September

Weight:

30%

Individual Work

All work is individual. You may discuss ideas, approaches and problems, but you should write every line of code yourself except for code copied from the lecture notes, lecture code or lab code. You **MUST NOT** let another student see your solution code, and you MUST NOT look at another student's solution code. More information about Academic Misconduct can be found at:

http://www.gsu.uts.edu.au/rules/student/section-16.html

Overview

Computer Store – A Computer Builder System

A friend of yours is setting up a computer building business, to help customers decide on what components they want in their system. Your friend has cajoled you, with the promise of cold pizza and fat coke, into writing some software to help this process.

The aim is to increase profits and makes their products more accessible to their customer base.

This requirement is urgent, as competitor have heard the rumors of this state-of-the-art system and they have also started working on similar prototypes.

Specification

The system should keep a catalogue of available parts, which can be displayed to the user, updated and filtered by type or price (or both).

The system should also allow the user to build a computer (more accurately a parts list) from parts in the catalogue, add and remove parts from the build, clear the build, and see the build (including the total cost).

The interface for the prototype will be a simple text interface, where parts are identified by their position in the catalogue/build.

As an advanced feature, your friend would like the user to be able to add several parts in one go.

Your friend has provided you with a nicely typed out text demonstration of how the system should work, along with some tests (… yet did not just code it…) to help.

An aside

While reading the first part of the specification, you will notice there is a lot going on.

- How many functions did you identify?
- How many classes did you identify?
- What are the fields in each class?
- How many goals did you identify?
- How many patterns did you think of that might be applicable?

This assignment will be challenging and you will probably want to manage your time well.

- How long do you think it will take you to code the functions?
- How long do you think it will take you to code each goal?

A good rule of thumb is to think of an estimate, and then multiply that number by 3 or 4!

To manage your time well, you may need to figure out which parts of the assignment you can start early.

- Which parts can you start now?
- Which parts can you start in week 6?

If you complete parts in the same week that you learn the topics (while they are fresh in your mind), they will take less time to complete.

The User Interface

Below is a sample I/O trace. The green text indicates user input (you are not expected to print color text). Not every conceivable scenario is shown below and you should submit your code to PLATE to see what specific scenarios are tested. You should also implement your solution in the same order as PLATE's test cases so that you can receive incremental feedback and marks as you progress.

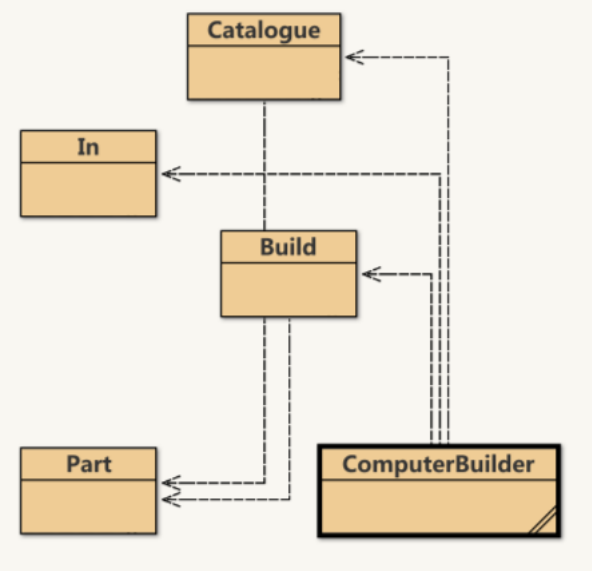
```
Welcome to Jaime's Computer Store
Quality Parts at the Best Prices
================================
1. Catalogue Menu
2. Build Menu
3. Exit
Select option: 1
Welcome to the parts catalogue.
Enter choice (a/r/s/f/x): s
1. STORAGE: evo 860 @ $155.00
2. KEYBOARD: daskeyboard @ $239.00
3. CPU: i5 @ $365.00
4. MEMORY: corsair 16G @ $299.00
5. MOTHERBOARD: ASUS ROG @ $159.00
6. CASE: sheetmetal box @ $39.00
7. CPU: Ryzen 7 @ $299.00
Enter choice (a/r/s/f/x): f
Enter type of part to view ('all' for no filtering): cpu
Enter minimum price ('-1' for no filtering): -1
3. CPU: i5 @ $365.00
7. CPU: Ryzen 7 @ $299.00
Enter choice (a/r/s/f/x): a
Enter part name: coolermaster junk
Enter part type: case
Enter part price: 50.00
Enter choice (a/r/s/f/x): s
1. STORAGE: evo 860 @ $155.00
2. KEYBOARD: daskeyboard @ $239.00
3. CPU: i5 @ $365.00
```

```
4. MEMORY: corsair 16G @ $299.00
5. MOTHERBOARD: ASUS ROG @ $159.00
6. CASE: sheetmetal box @ $39.00
7. CPU: Ryzen 7 @ $299.00
8. CASE: coolermaster junk @ $50.00
Enter choice (a/r/s/f/x): x
1. Catalogue Menu
2. Build Menu
3. Exit
Select option: 2
Let's build a 1337 box!
Enter choice (n/a/r/v/c/x): a
Enter catalogue number of the part: 3
Enter choice (n/a/r/v/c/x): a
Enter catalogue number of the part: 1
Enter choice (n/a/r/v/c/x): a
Enter catalogue number of the part: 2
Enter choice (n/a/r/v/c/x): a
Enter catalogue number of the part: 4
Enter choice (n/a/r/v/c/x): a
Enter catalogue number of the part: 5
Enter choice (n/a/r/v/c/x): a
Enter catalogue number of the part: 6
Enter choice (n/a/r/v/c/x): c
The build is functional.
Enter choice (n/a/r/v/c/x): v
1. CPU: i5 @ $365.00
2. STORAGE: evo 860 @ $155.00
3. KEYBOARD: daskeyboard @ $239.00
4. MEMORY: corsair 16G @ $299.00
5. MOTHERBOARD: ASUS ROG @ $159.00
6. CASE: sheetmetal box @ $39.00
Total Price: $1256.00
Enter choice (n/a/r/v/c/x): r
Enter number of part to remove: 1
Enter choice (n/a/r/v/c/x): a
Enter catalogue number of the part: 7
Enter choice (n/a/r/v/c/x): v
1. STORAGE: evo 860 @ $155.00
2. KEYBOARD: daskeyboard @ $239.00
3. MEMORY: corsair 16G @ $299.00
4. MOTHERBOARD: ASUS ROG @ $159.00
5. CASE: sheetmetal box @ $39.00
6. CPU: Ryzen 7 @ $299.00
Total Price: $1190.00
Enter choice (n/a/r/v/c/x): c
The build is functional.
Enter choice (n/a/r/v/c/x): x
1. Catalogue Menu
2. Build Menu
3. Exit
Select option: 3
```

Requirements

- Your design will consist of exactly the following classes with the listed fields, declared as indicated. You may not add or remove classes or fields; however, you may add constructors, functions and procedures to complete your design (in fact, you will have to!). You should pay careful attention to the tests on PLATE, as these will help guide you with some (but not all) of these methods.

- To help visualize the design, a partial class diagram has been provided



- Classes – your design will consist of these 5 classes:

  1. ComputerBuilder
  2. Build
  3. Catalogue
  4. Part
  5. In (this is just the class you've been using throughout the labs to facilitate simpler I/O – just copy it over)

- Fields – the classes will have the following fields:

```
public class ComputerBuilder {

    private Catalogue catalogue;

    private Build currentBuild;

}

public class Build {

    private List<Part> parts;

}

public class Catalogue {

    private List<Part> parts;
```

```
}

public class Part {

    private String name;

    private String type;

    private double price;

}
```

- The fields also have some additional requirements and structures:

  Lists all have the abstract type of List<>, but must be instantiated with a concrete type that implements the List<> behavior (you will see two of these in week 6, you can choose either – you may also want to think about why you might do things this way).

  The type String in Part is stored in lowercase.

- Constructors – the constructors of the class have the following requirements:

  1. All constructors initialize the fields of their class.
  2. The ComputerBuilder constructor takes no parameters.
  3. The Catalogue constructor takes no parameters, but will add some initial Parts to the catalogue:

Name	Type	Price
evo 860	storage	155.00
daskeyboard	keyboard	239.00
i5	cpu	365.00
Corsair 16G	memory	299.00
ASUS ROG	motherboard	159.00
sheetmetal box	case	39.00
Ryzen 7	cpu	299.00

  4. The Build constructor takes no parameters.
  5. The Part constructor takes three parameters, corresponding to the name, type and price, with the same types as the respective fields.

- toString() – Catalogue, Build and Part will each have a toString() function, see the I/O trace and tests for the formats.

- The main method of the program will be in the ComputerBuilder class.

Expected Workload

The time to do the assignment to a credit/distinction level has been estimated at 25 hours for a student of average ability who has completed all the tutorial and lab exercises.

Online Support

The Assignment 1 discussion board has been set up on UTSOnline so that students can ask questions, and other students can reply. The course coordinator will only post a reply only if the student response was wrong, or in the case of correcting a mistake in the assignment specification.

You must not post or share Java code to the discussion board. The board is there to help you, not to provide the solution. **Posting your code is academic misconduct and will reported**. Each time this rule is violated, the code will be removed and replaced with a comment of the form: “Strike 1: Posting code”. After 3 strikes, the discussion board will be deleted because it did not work.

FAQs (Frequently Asked Questions) and their answers will be posted on PLATE alongside the assignment specification. If you have a question, check the FAQ first; it may already be answered there. You should read the FAQ at least once before you hand in your solution, but to be safe check it every couple of days. Anything posted on the FAQ is considered to be part of the assignment specification. The FAQ will be frozen (no new entries) two days before the due date; no questions will be answered after it is frozen.

If anything about the specification is unclear or inconsistent, contact the subject coordinator who will try to make it clearer by replying to you directly and posting the common questions and answers to the FAQ. This is similar to working on the job, where you ask your client if you are unsure what has to be done, but then you write all the code to do the task. Email huan.huo@uts.edu.au to ask for any clarifications or corrections to the assignment.

PLATE Marking

Your solution is marked for correctness by PLATE (https://plate.it.uts.edu.au/) by comparing the output of your system to the output of the benchmark system. You can submit a solution to PLATE many times; I urge you to do this, so you receive credit for your work. Submission takes the same form as for the labs, you must package your assignment in a JAR file, including the source code.

PLATE will test the features of your program in a certain order: Classes and fields, then constructors, then goals from basic to advanced. PLATE cannot test the more advanced goals until the basic goals are working. To receive marks, you must pass PLATE’s test cases in the order in which PLATE tests them.

Your code is marked by software, so you can get a good mark by fooling or spoofing the software. If you spoof a task worth N marks, you receive a penalty of 2*N marks.

Submission and Return

Your solution is to be submitted to PLATE at https://plate.it.uts.edu.au/ under Applications Programming / Assessments / Assignment 1, in the same manner as your labs – package in a JAR file including the source code. Your provisional mark and feedback is generated immediately each time you submit to PLATE. However, analysis of spoofing, plagiarism, collusion and general cheating is done in the two weeks following the due date. If you are suspected of Academic Misconduct, I will forward your case to the Misconduct Committee and will notify you by email.

There is no scheduled late submission period. An extension of up to one week may be given by the subject coordinator before the due date; you have to supply documentary evidence of your claim. An extension CANNOT be given after the due date.

You may also apply for special consideration for reasons including unexpected health, family or work problems. More information about how to apply for special consideration can be found at:

http://www.sau.uts.edu.au/assessment/consideration.html.

Marking Scheme

The marks for the assignment are divided into the following functionality components (note that individual tests may test several functionality components, and a functionality component may be tested by several tests):

Functionality Component	Mark Allocation
Fields	7
Constructors	7
Main Menu	5
Catalogue Menu	5
Build Menu	5
Add Part to Catalogue	5
Remove Part from Catalogue	8
Filter by Type	7
Filter by Price	7
Filter by Type and Price	7
Add Part to Build	6
Remove Part from Build	6
Check Validity of Build	10
Clear Build/Start New Build	5
Add Several Parts to Build at Once	10

This adds to a mark out of 100, at makes up 30% of your final assessment mark.