

# AP<sup>®</sup> Computer Science (A)

## 1st Semester Final Project

### Bedroom Object Interaction

### A & B Version

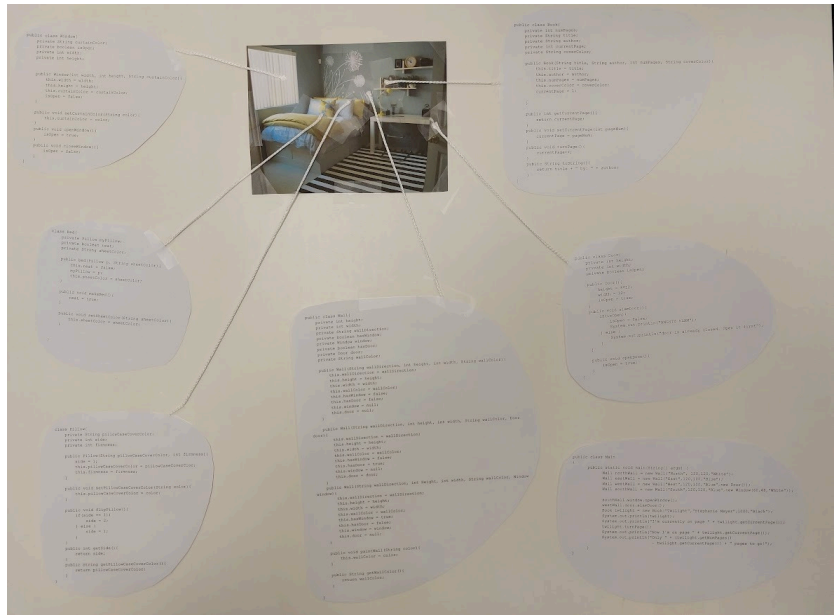
#### Assignment Purpose:

The purpose of this program is to create a properly encapsulated Bedroom class with multiple desired bedroom objects, each object class contains instance variables and methods with data structure Arrays and ArrayLists, and demonstrating proper Class interaction with data structures (Array, 2D Array, ArrayList)

Your final project this semester is to create an Object-Oriented version of your dream bedroom. Choose at least 6 "things" in your room or your dream bedroom (bed, wall, windows, door, tv, phone charger, chair, desk, book, poster, mini-fridge, hamster cage, closet, etc). On this google doc, create class diagrams for each Class that includes 3-4 instance variables and methods for each Object.

You could simply choose your own bedroom and take a picture. Or pick the dream bedroom from internet pictures or movie shows, for example, Zach Morris from Saved by the Bell, Hannah Montana's bedroom, or the Brady Bunch Boys bedroom. Make sure the items from your 2D Array and ArrayList are visible and reasonable.

See below for an example of a bedroom picture and the poster made as the end product. When you are doing Level B check ins with Ms. Xu, she could notify you if your dream bedroom was identical to another student, you would have to choose a different bedroom picture. The picture of your bedroom must be UNIQUE!



## Your project should include the following:

### Constructors:

- For at least one of the objects, include a default constructor with no parameters
- For at least one of the objects, include a constructor with parameters for all instance variables
- For at least one of the objects, include multiple constructors with different parameters (remember the rule about matching data types)

### Instance Variables (attributes):

- Each object should have at least 3 private instance variables that are initialized in a constructor.
- Use a variety of data types for your instance variables including int, double, boolean, and String
- For at least one of your objects, make an instance variable type one of the other objects (or the object type itself if you want to do a comparison method)
- At least one one-dimensional array and one two-dimensional array contains **objects** ( 2D array of books, shoes, baseball cards, etc...)
- Level A: At least one ArrayList contains **objects** (a list of books, Thomas the Tank Engines, a list of things to do on paper, etc.). The ArrayList concept is from Chapter 7.

### Methods:

- For each object, include at least 2 accessor/mutator (get/set) methods for the instance variables (be sure you are using void / return methods correctly and returning the correct data types as necessary)
- For at least 2 of the mutator methods, include some kind of a check that prevents an unintended consequence (prevent dividing by zero, setting a number to a negative value when it shouldn't be, etc)
- For at least 3 of the objects, include a method outside a simple "get/set" that will "do something" besides just returning the current value of an instance variable - example: Book.turnPage() or Door.slamDoor();
- For at least 3 of the objects, include a toString method that will return a String description of the object
- At least 1 method (non constructor) that requires you to pass either 1D array or 2D array from outside the class, it will update your 2D array instance variables (for example, you borrowed more books from LRC, and brought them back to your bedroom bookcase which contains 2D array of books). Must display your knowledge of array traversal with for loops. (traditional and for each loop)

### Main Method:

- Create a main method where you instantiate objects of your different classes. Then test the various object methods to verify they are working correctly. Your code output shows the correct value printed.

### Poster for Gallery Walk (After completion of Version A):

When finished, you will take a picture of your room or dream bedroom (be sure it's school appropriate), print out the various classes (10 point Courier New font), and glue down the classes and picture to a poster board. Attach yarn between each Class file and where the object is in the picture. This is the first semester grade!!!

-----

#### Step 1

List the objects you will create

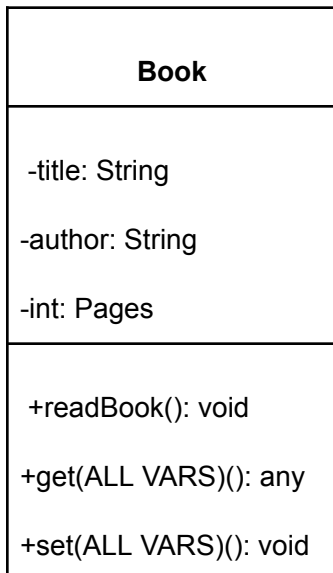
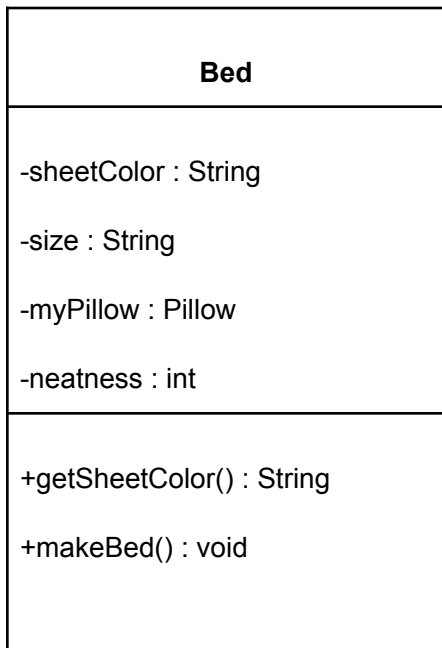
- |              |             |
|--------------|-------------|
| 1) Bed       | 5) Desk     |
| 2) Bookshelf | 6) Computer |
| 3) Chair     | 7) iPhone   |
| 4) Closet    | 8) Monitor  |

## 9) Book

## 10) Shoe

Step 2: Create a UML diagram for each object. If you end up needing to make a change to a method or add an attribute later, that's okay, but this should be your guiding document for the rest of the project.

Examples: (example – edit it as necessary)



Pants
<div>-size: float</div> <div>-color: java.awt.Color</div> <div>-brand: String</div>
<div>+wearPants(): void</div> <div>+get(ALL VARS()): any</div> <div>+set(ALL VARS()): void</div>

Shirt
<div>-size: float</div> <div>-color: java.awt.Color</div> <div>-brand: String</div>
<div>+wearShirt(): void</div> <div>+get(ALL VARS()): any</div> <div>+set(ALL VARS()): void</div>

Bed
-size: float -color: java.awt.Color -model: String
+sleep(): void +get(ALL VARS)(): any +set(ALL VARS)(): void

Chair
-size: float -color: java.awt.Color -model: String
-sit(): void +get(ALL VARS)(): any +set(ALL VARS)(): void

Closet
-pants: Pants[] -shirts: Shirt[] -shoes: Shoe[]
+get(ALL VARS)(): any +set(ALL VARS)(): void

Shoe
<div>-size: float</div> <div>-color: java.awt.Color</div> <div>-brand: String</div>
<div>+wearShoe(): void</div> <div>+get(ALL VARS)(): any</div> <div>+set(ALL VARS)(): void</div>

Computer
<div>-appNames: ArrayList&lt;String&gt;</div> <div>-model: String</div> <div>-color: java.awt.Color</div>
<div>+useApp(): void</div> <div>+get(ALL VARS)(): any</div> <div>+set(ALL VARS)(): void</div>

IPhone
-appNames: ArrayList<String> -model: String -color: java.awt.Color
+useApp(): void +get(ALL VARS)(): any +set(ALL VARS)(): void

Step 3: Start coding based on the UML design. You should have Bedroom Class and 6 other classes representing Bedroom “things” of your choice.

Step 4: Coding for the main.java or tester to test out all object/instance variables/methods to prove that they are working correctly

Step 5: Make the poster for Gallery Walk in class. Follow the instructions.

Step 6: Complete the reflection below.

## **Level B Version Requirements (Check-In with Ms. Xu Due before the last day of Final Exam)**

- Have at least 4 object classes ready that work together inside your bedroom class.  
Your bedroom class or object class must contain 1D Array and 2D Arrays of objects
- All is working well and being fully tested
- It will give you up to 85% of the credit for this project based on the above requirements. 20 Points

## **Level A Version Requirements (Due on the 1st day after Winter Break)**

- Have at least 6 object classes that work together in your bedroom class
- Your bedroom class must contain 1D array, 2D array and ArrayList
- All is working well and being fully tested.
- Poster is completed with pictures of the bedroom and code structure is displayed clearly
- Reflection document (including the source code, video links, pictures of poster, etc.) is submitted to Schoology
- It will give you up to 105% of the credit for this project

## GRADING PROCEDURE:

- 1) You will submit THIS document with all your answers, code and screenshots.
- 2) Make sure you have **thoroughly** tested your methods with Array and ArrayList yourself.
- 3) Make sure your name is listed as an author in each Java class.
- 4) Paste the your own testing results with the Bedroom class code
- 5) Make a 5-minute video to talk about important features of your code (your own choice), attached the link
- 6) Complete the reflection on what you have learned and how you have fixed the issues
- 7) Submit this document on Schoology
- 8) Bring the poster to School for a gallery walk presentation.

## Fill in information below

### Your Name:

Christopher Mustard

## Copy/Paste your test snapshots below:

```
PS C:\Users\CJMus\OneDrive\Desktop\APCSA Bedroom Project> cd 'c:\Users\CJMus\OneDrive\Desktop\APCSA Bedroom Project'; & 'C:\Program Files\Eclipse Adoptium\jdk-21.0.5.11-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\CJMus\AppData\Roaming\Code\User\workspaceStorage\9cfc44cccfc728efc6487aadd6b33011\redhat_java\jdt_ws\APCSA Bedroom Project_7d1fdb03\bin' 'Bedroom'
Desk {
  phone=IPhone {appNames=[Roblox, VS-Code, Google Classroom, Fortnite], model='15 Pro Max', color=java.awt.Color[r=0,g=0,b=0]}
  chair=Chair {color=java.awt.Color[r=0,g=0,b=0], size=2, model='Walker'}
  3 computers
  3 monitors
}
Bookshelf {
  numberOfShelves=8
  color=java.awt.Color[r=0,g=0,b=0]
  books=8 shelves of books
}
Closet {
  pants=2 pairs of pants
  shirts=2 shirts
  shoes=1 pairs of shoes
}
Bed {
  color=java.awt.Color[r=135,g=206,b=250]
  size=14
  model='Purple'
}
```

Please insert the link to all source codes (must be shared) or copy/paste the original code here if you use BlueJ:

All java files ends with .java(BlueJ) || Main.java(Repl))

../src/

Your explanation in video here (must be shared, key terms must be worded out and explained with details, no more than 5 minutes long):

./explanation.mp4

Please enter the resource from your code!!! (links to the idea or code you have borrowed)

N/A

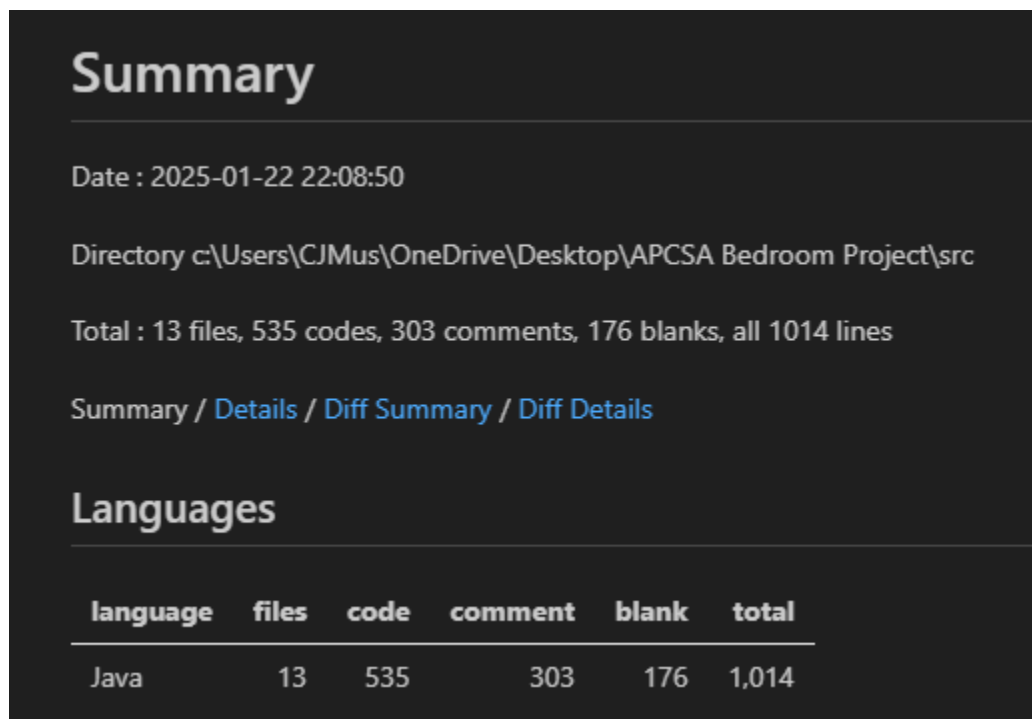


Type in your reflection here, for example, did you run into any issues/bugs, how did you solve them? What did you learn from this project, any suggestions for making this project better?

My most considerable struggle with this project was the sheer size. I completed this project in around 30 minutes, though it was repetitive.

Do you deserve the extra credits? Why?

I spent quite a while implementing everything we covered in our classes, sticking to the project guidelines. I have provided advanced concepts while also expanding the depth of my project.

A screenshot of a code summary tool interface with a dark background. It shows a 'Summary' section with a date, directory path, and file statistics. Below that is a 'Languages' section with a table of code statistics for Java.

Summary					
Date : 2025-01-22 22:08:50					
Directory c:\Users\CJMus\OneDrive\Desktop\APCSA Bedroom Project\src					
Total : 13 files, 535 codes, 303 comments, 176 blanks, all 1014 lines					
Summary / <a href="#">Details</a> / <a href="#">Diff Summary</a> / <a href="#">Diff Details</a>					
Languages					
language	files	code	comment	blank	total
Java	13	535	303	176	1,014

Copy/Paste your dream bedroom picture poster below:

./Class Trace.png