
Important Guidelines

All labs will be submitted on Blackboard, unless otherwise noted in the lab description. Export the project and upload the zip file. **In Eclipse, click on the project, select Export, and choose Archive File.**

You need to submit a zipped file that contains the whole project. You need to submit a zipped version of your whole project. If you submit only the class files or src files, that is not acceptable.

We will then import your project and run from our end. If you don't submit the whole project, we won't be able to run your project.

The name of each project must be **abc123-labX**, where abc123 is your UTSA ID and X is the number of the lab. All letters are lowercase.

The name of your submission must be **abc123-labX.zip**. All letters are lowercase.

Any submission not following these requirements may not receive credit.

Reminder: laboratories must be completed individually. Collaboration with anyone violates the policies for academic integrity.

You may discuss algorithms and approaches with others. You may not code together or share a computer, under any circumstance.

If you have questions or concerns about this policy, contact your instructor immediately.

In the rare event that a change must be made to an individual lab description, the change will be announced in lecture, and a note will appear on the lab submission on Blackboard.

Lab 1

Objectives:

- Familiarity with Eclipse IDE
- Java syntax, the String class
- Object oriented programming

Task: Just helping the community!

One of the elementary school's principal Ms. Emily Johnson has asked for some basic software to help manage her school. Version 1.0 of the software will create Java objects to represent the school itself, the zones/areas within the school, and the students in each zone. With this software, school administrators will be able to plan and take care of the students in each area of the school more efficiently as opens for the school year. For example, when the students are at the cafeteria, it will show their food preferences (veg or non-veg). When the students are at playground, it will show if the student needs any kind of special attention or accommodation etc. When the students are at library, it will show the information related their reading habit/preferences, literacy etc.

We will be only focusing on the food preference part for the Version 1.0.

Getting Started

Create a new project in Eclipse, following the lab guidelines. Create the following new Java classes, in the default package: (*Use these names exactly!*)

- School.java
- Zone.java
- Student.java

Lab1.java

Place the Lab1.java class provided here in the same default package within your project.

Note that this code will not compile until you have completed the requirements of this lab. There will syntax errors until all dependencies (classes and methods) are

implemented.

Lab1.java has a main method and will be the class to run our application. Follow the remaining instructions for each class in this lab in order to get your code to compile - **do not change the given class.**

School

This class will represent a School object, which we will define as having:

- A name, represented as a String (i.e.: Monroe May Elementary School)
- Zones, stored as an array of Zone objects (i.e. Zone[])
- A toString() method, which calls upon the toString() method in Zone.java to return as a String all needed information.
- An addZone(..) method, which takes as a parameter a Zone object and returns nothing.

This class must have a constructor and getters and setters to accommodate its variables.

Zone

This class will represent a Zone object, which we will define as having:

- A name, represented as a String (i.e.: Cafeteria Zone)
- An array of Student objects.
- A toString() method which returns a String representation of all students in the zone
- An addStudent(..) method, which takes as a parameter a Student object and returns nothing.

This class must have a constructor and getters and setters to accommodate its variables.

Student

This class will represent a Student object, which we will define as having:

- A name, represented as a String (i.e. Lisa Lawrence)

- A level, represented as a String (i.e. Pre-K)
- Whether or not the Student is a vegetarian - very important info for the school kitchen ! This will be represented as a boolean (i.e. *false*, for Lisa Lawrence)
- A toString() method which returns a String representation of the student.

This class must have a constructor and getters and setters to accommodate its variables.

As this lab is meant to review regular arrays in Java, no other data structure may be used to store the objects required. (No ArrayLists are permitted, for example).

Output

Once your code compiles, you will be able to examine the output of your program. The output of your program **must** match the format of the sample below. This sample is the result of running the Lab1.java class with the given main method.

```
Welcome to Monroe_May_Elemn_School!
-----
Cafeteria Zone:
* Lisa_Lawrence - Pre-K (non-veg)
* Oliver_Qu    - K (non-veg)
* Oscar_Smith  - K (non-veg)

Playground Zone:
* Freddie_Miller - 1st (non-veg)

Library Zone:
* Ada_Wilson - 2nd (veg)
* Jasmine_Nadia - 3rd (veg)
* Tanya_Henson - 4th (veg)
```

Rubric:

- (10pts) Comments & Formatting - All code is properly formatted and commented. This includes your full name and UTSA ID (abc123) at the top of each class. Each class and method should have a description of what it does,

and methods should additionally have a description of any parameters and returned data, as applicable.

- (10pts) Submission - The Eclipse project is correctly submitted to Blackboard, as abc123-lab1.zip.
- (50pts) Correctness - All classes are declared exactly as described above and function appropriately.
- (30pts) Tests - The grader will test your code by modifying the main method in Lab1.java to add additional zones and students. If coded according to the requirements of this lab, your submission will output the correct result. *Note however if you "hard code" any portion of the lab (except Lab1.java, which you should not modify) your submission will fail this test case.)*

```
package Lab1;

public class Lab1 {

    public static void main(String[] args) {
        School elem = new School( "Monroe_May_Elemn_School", 7 );

        Zone r = new Zone( "Cafeteria", 10 );
        Zone t = new Zone( "Playground", 2 );
        Zone b = new Zone( "Library", 20 );

        Student bl = new Student( "Lisa_Lawrence", "Pre-K", false );
        Student de = new Student( "Oliver_Qu", "K", false );
        Student ec = new Student( "Oscar_Smith", "K", false );
        Student re = new Student( "Freddie_Miller", "1st", false );
        Student lt = new Student( "Ada_Wilson", "2nd", true );
        Student sp = new Student( "Jasmine_Nadia", "3rd", true );
        Student bb = new Student( "Tanya_Henson", "4th", true );
```

```
r.addStudent( bl );  
r.addStudent( de );  
r.addStudent( ec );  
t.addStudent( re );  
b.addStudent( lt );  
b.addStudent( sp );  
b.addStudent( bb );  
  
elem.addZone( r );  
elem.addZone( t );  
elem.addZone( b );  
  
System.out.println( elem );
```

```
}
```

```
}
```
