

This Week's Objectives:

Students will be able to:

1. explain the requirements and constraints of the course using the syllabus
2. recognize that accepting the syllabus and continuing the course represents their commitment to following all the stated requirements of the syllabus
3. read, understand, and implement program standards
4. configure and apply a documentation standard (Javadocs) in code
5. implement array management in Java, including creating, adding, accessing, and removing data, and reporting array state components
6. analyze array actions, including
7. explaining the actions and/or predicting the results of a given code segment
8. identifying issues that might cause runtime errors and/or data corruption
9. develop and test a basic array-based wrapper class, including implementing specified components
10. know and be able to define, apply, and/or explain the following terms:
11. encapsulation, data hiding, loose- and tight- coupling (as pertains to subroutines/methods)
12. precondition, postcondition, exceptional condition, error condition
13. object-oriented programming (OOP), and how it is the same or different from procedural programming
14. abstract data types (ADT), and how they are the same or different from data structures

This Week's Assignment(s):

1. Read the objectives carefully, make sure you can answer any question provided (before class)
2. Make sure you understand what each is asking.
3. Use references as needed, including the textbook, to respond to each objective as needed.
4. Finally, make sure you can meet all the objectives by the end of the week.

If you can't, come see one of us for help

You are responsible for all of the objectives throughout the course; they are cumulative

All quizzes and examinations will be a part of, or directly related to, the stated objectives

You may be quizzed or tested on any stated objective from the week it was introduced until the end of the semester

It would be a good idea to make and keep notes for all of the objectives you have studied

Next Week's Objectives (Tentative, Subject to Modification):

[In addition to and including all the previous weeks' objectives]

ASSIGNMENT REQUIREMENTS

This assignment is due by 6:00 pm on the date specified as the title of, and at the beginning of, this assignment page. Any assignments turned in after 6:00 pm but received before 11:59 pm on the same date will be accepted with a 50% reduction of the graded score. No assignments will be accepted at or after 12:00 midnight of the due date. Note that the Blackboard Learn clock will be considered the standard; you would be wise to turn your assignments in well before these specified deadline times. Technical and/or time-related issues related to uploading assignments are not likely to be accepted as reasons to recover credit.

ASSIGNMENT OBJECTIVES

Students will review the development and management of a one-dimensional array within a class wrapper

Students will review use of Java arrays, and practice basic algorithmic activities with arrays

Students will review the pass-by-copy practice of Java, along with managing potential aliasing issues in method development

Students will practice development, implementation, and usage of supporting/utility methods; students will not use Java utilities such as length, size, etc., unless they are specifically permitted

Students will generate classes that manage data in the form of a one-dimensional array

Students will practice implementing methods in such a way that code is rarely, if ever, repeated by implementing supporting or utility methods in their development activities

Students will demonstrate analytic and diagnostic competence with testing and verifying all components of the given assignment

ASSIGNMENT OVERVIEW

Students will develop a class wrapper that provides the tools of a one-dimensional array with protected boundaries

ASSIGNMENT SPECIFICATIONS

Students must develop the code to manage a one-dimensional array with accessing and modifying tools

Students must develop other supporting methods as specified (below) to manage the wrapper operations

Students will also document all classes and methods using the Javadoc commenting process; comments are not required to be exactly the same as found in the supporting document (below) but they should be semantically equivalent

Students will upload the ArrayClass.java file to this assignment; any other uploaded files will result in a reduction of the project grade, and in some cases, may cause a complete loss of credit (e.g., uploading *.class, or other unusable files)

Other Constraints:

The General Usage Rubric will be applied to all code developed by the students; verify that your code does not incur any credit reduction due to these standards

All methods must be placed in the class file in alphabetical order so methods can be found by graders; methods not found will not be graded

Students are given a maximum of three attempts to upload their work to this BBLearn assignment page

ASSIGNMENT DOCUMENTS

Assignment Specification (Javadoc output for the project):

https://www.cefns.nau.edu/~mel346/Project_1/

Project Rubric: [here](#)

Rubric - PA01

- Program builds correctly, no errors or warnings: +5

- Methods are written correctly: +40
 - All constructors (+4)
 - accessItemAt (+4)
 - appendItem (+4)
 - clear (+4)
 - getCurrentCap (+4)

- `getCurrentSize` (+4)
 - `insertItemAt` (+4)
 - `isEmpty`, `isFull` (+4)
 - `removeItemAt` (+4)
 - `resize` (+4)
- All other specifications followed: +5
 - correct file upload
 - correct method names, variable names, etc.
 - appropriate use of constant (final) value
- Appropriate Javadoc documentation with each method: +5
 - `@param`, `@return` correctly used
 - notes with `<p>` correctly used as dividers
 - Javadoc output is not required at this point
- General Usage Rubric reductions
 - see GUR document

Organize your files under `p1_package` (this is a requirement)