Background

This project involves implementing a **R**e-sizable **D**ynamic **Array** (RD_Array) Abstract Data Type (ADT). The class declaration is available on-line for download. There is also a lot of other given code available to copy from the on-line documentation. We shall be replicating much of the functionality of the vector template class, just with a different name.

Documentation for this class and its methods is available for download on Canvas and review at:
http://people.cs.georgetown.edu/~addison/projects/fall2024/p5docs/index.html

From the on-line documentation you can view the full C++ code for all .h files. You can also copy and paste the code into your project. Like previous projects, you should do that first. All given code may be used in full, or in part, as your own without attribution.

You must thoroughly test your implementation of the RD_Array<T> class in main.cpp. Instantiate an RD_Array<char> object and call ALL of its member functions to demonstrate they are correctly implemented.

You may not modify the RD_Array class name or the public interface. As before evaluation of your project will include an Autograder evaluation and a review by one of the Teaching Assistants. The Autograder driver program will only include main.h. Put using namespace std; and all necessary preprocessor directives in your main.h. You should not need to declare any additional methods or data members. Use the provided Makefile to compile your program and prepare the final compressed file for submission.

Getting Started

Set up a new project with your applicable files from Project 4 as the starting point. Next add new project files and copy and paste the given class declarations and other code. Once that is done, I recommend that you immediately write function stubs for **all** methods of **all** new classes. Thereafter, use stepwise refinement and incremental development to implement the new code for this project.

Driver Program

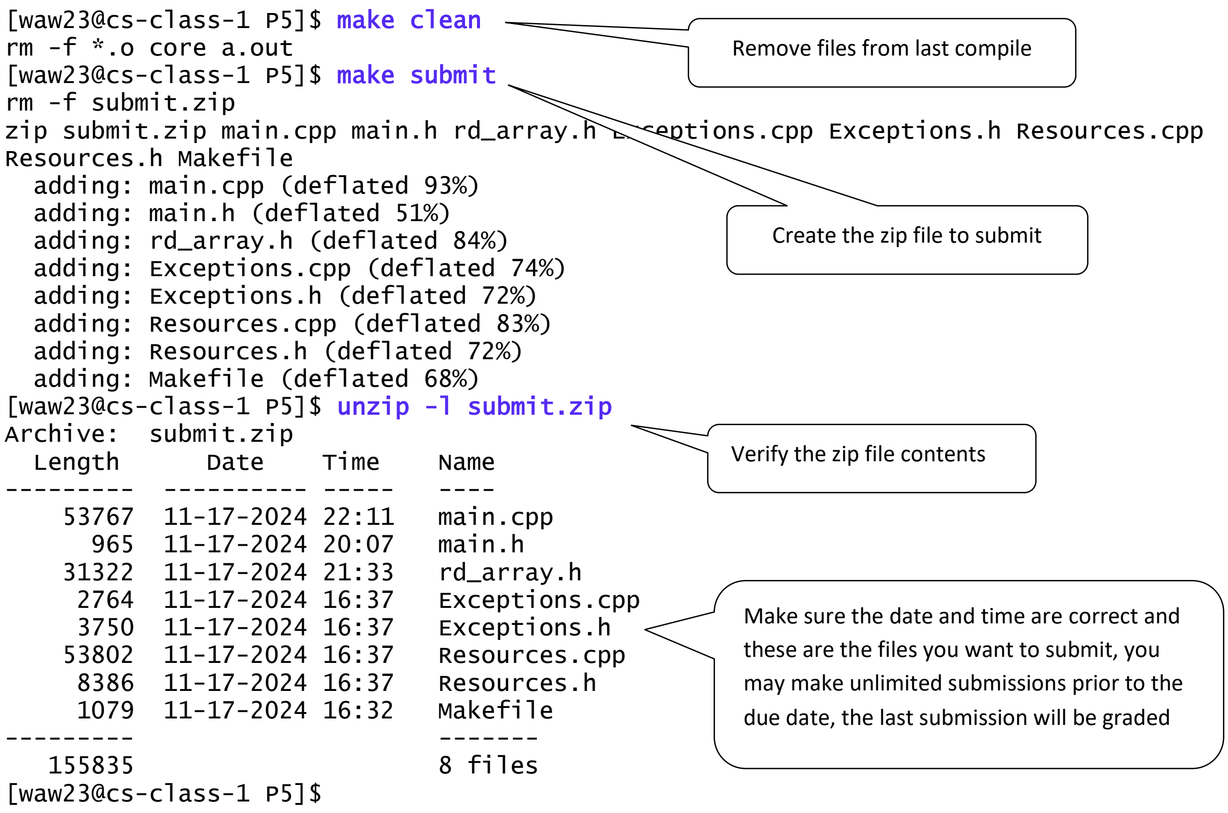
For this project, the driver program (main.cpp) does not need to have any specific code. Some example test code may be provided that you may use as your own in the driver program. If that is done, you should still add more test code as necessary to ensure that all class member functions are working correctly.

Submission Details

What to submit: One compressed file containing all source code and the `Makefile`, and the `.pdf` file containing experimental results, associated with this project. The file name must be `submit.zip`. You must separate your class specification details from your class implementation details. Therefore, you must prepare a specification file (`<filename>.h`) and an implementation file (`<filename.cpp>`) for each set of related classes. The file `rd_array.h` is different. Since it contains template classes, the class declarations and all member function implementation code shall also be stored in the `.h` file. **There is no `rd_array.cpp` file for this project.** Ensure that your `.h` files contain sufficient comments for each data member and class method. Additionally, you must provide file `main.cpp` (that contains function `main`) along with its associate `main.h` file. This "driver" code is where class objects are instantiated and functionality of the software is demonstrated. Use the following file names (**with spelling and capitalization exactly as shown**):

`main.h, main.cpp`
`rd_array.h` (copy and paste class declaration, function stubs, and given code from the on-line documentation)
`Resources.h, Resources.cpp, Exceptions.h, Exceptions.cpp` (reuse your files from Project 4)
`Makefile` (given file)

Creating `submit.zip`*:* **Please, PLEASE, PLEASE** use the provided `Makefile` and create your `submit.zip` file on the class server. If you create the compressed file on your laptop it is highly likely something will go wrong even though it looks fine. It is easy to compress links to files, instead of actual files. It is easy to have the folder containing the project files included in the compressed file. If anything such as that happens; your program will not compile, automated grading programs will fail, **and YOU WILL GET A ZERO for the project**. Assuming all of your files are in the same folder on the server, the process to create the `submit.zip` file is shown below.



```
[waw23@cs-class-1 P5]$ make clean
rm -f *.o core a.out
[waw23@cs-class-1 P5]$ make submit
rm -f submit.zip
zip submit.zip main.cpp main.h rd_array.h Exceptions.cpp Exceptions.h Resources.cpp
Resources.h Makefile
  adding: main.cpp (deflated 93%)
  adding: main.h (deflated 51%)
  adding: rd_array.h (deflated 84%)
  adding: Exceptions.cpp (deflated 74%)
  adding: Exceptions.h (deflated 72%)
  adding: Resources.cpp (deflated 83%)
  adding: Resources.h (deflated 72%)
  adding: Makefile (deflated 68%)
[waw23@cs-class-1 P5]$ unzip -l submit.zip
Archive:  submit.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
    53767  11-17-2024 22:11   main.cpp
      965  11-17-2024 20:07   main.h
    31322  11-17-2024 21:33   rd_array.h
     2764  11-17-2024 16:37   Exceptions.cpp
     3750  11-17-2024 16:37   Exceptions.h
    53802  11-17-2024 16:37   Resources.cpp
     8386  11-17-2024 16:37   Resources.h
     1079  11-17-2024 16:32   Makefile
---------                     -------
   155835                     8 files
[waw23@cs-class-1 P5]$
```

Remove files from last compile
Create the zip file to submit
Verify the zip file contents
Make sure the date and time are correct and these are the files you want to submit, you may make unlimited submissions prior to the due date, the last submission will be graded

Grading

This graded assignment is worth 100 points and will be counted as part of the *Programming Projects* category for the course. Your final score is based on automated tests, as well as a manual review conducted by one of our Teaching Assistants (TAs). A detailed rubric of points and a list of common deductions will be published separately.

Due date/time: No later than end-of-day (11:59pm) on the due date for your section listed on Canvas. Late submissions will be penalized up to 1% for each minute late. If late points result in a zero, you may still turn in the project to receive feedback but the grade will remain zero. In general requests for extensions will not be considered. "Do overs" and/or resubmissions will not be allowed. **Double check the files you submit to Canvas:** Download the actual `submit.zip` file that you submitted to Canvas. Store it in an empty folder in a location separate from your project on your local computer. Unzip the file and verify the contents are the most recent, correct files that you want graded. If not, fix the `submit.zip` file and resubmit. Repeat until you are sure the correct files have been submitted to the Assignment on Canvas.

Academic Integrity

This is an individual project and all work must be your own. Refer to the guidelines specified in the *Academic Honesty* section of this course syllabus or contact me if you have any questions.

Include the following comments (with appropriate substitutions) at the start of each file in your project:

```
/*
 * <FileName>.<file extension>
 *
 *
 *  COSC 1030 Fall 2024
 *  Project #5
 *
 *  Due on: DEC 6, 2024
 *  Author: <last name>  <netID>
 *
 *
 *  In accordance with the class policies and Georgetown's
 *  Honor Code, I certify that, with the exception of the
 *  class resources and those items noted below, I have neither
 *  given nor received any assistance on this project.
 *
 *  References not otherwise commented within the program source code.
 *  Note that you should not mention any help from the TAs, the professor,
 *  or any code taken from the class textbooks.
 *
 */
```

Course Materials Notice

The materials used in Georgetown University courses ("Course Materials") generally represent the intellectual property of course instructors which may not be disseminated or reproduced in any form for public distribution (e.g., sale, exchange, etc.) without the written permission of the course instructor. Course Materials include all written or electronic documents and materials, including syllabi, current and past examination questions/answers, and presentations such as lectures, videos, PowerPoints, etc., provided by a course instructor. Course Materials may only be used by students enrolled in the course for academic (course-related) purposes.

Published course readings (book chapters, articles, reports, etc.) available in Canvas are copyrighted material. These works are made available to students through licensed databases or fair use. They are protected by copyright law, and may not be further disseminated or reproduced in any form for distribution (e.g., uploading to websites, sale, exchange, etc.) without permission of the copyright owner.

More information about intellectual property and copyright can be found here:

https://library.georgetown.edu/copyright

More information about computer acceptable use policy and intellectual property can be found here:

https://security.georgetown.edu/acceptable-use/#