# CMSC131 Fall 2021 (https://www.cs.umd.edu/class/fall2021/cmsc131-01XX-03XX/)

## Project #2, Java Loops (Due Wednesday, Sep 22, 11:55 pm)

### Objectives

This project will allow you to practice Java's loops.

### Overview

In this project you will implement three programs: Convert (converts from decimal to octal), ThrowDie (simulates throwing a die), and Access (simulates accessing a system).

For this project you can work with classmates, but you may not exchange code. Make sure you add the name of classmates you work with to the top of the Convert.java file (in comments).

If you have any questions about the project, first check the first pinned message you will find in the piazza folder we have created for this project. This pinned message has clarifications we have made so far. In addition, read the project description again as you can find the answer in the description.

### Announcements

- Make sure you check the Piazza announcements folder often (at least twice a day in the morning and evening)
- Lecture examples and slides can be found in the class web page ("Schedule" section).
- **Make sure you have an alternative (e.g., a landline, an ipad, codes, etc.) for Duo (school) authentication. If you only rely on your phone for Duo authentication and you lose your phone, you will not be able to submit or access school resources. Have a backup plan in case you lose the primary device (e.g., phone) you use for authentication.**
- Nelson takes academic integrity seriously. Abide by the rules defined in the sylllabus.

### Grading

- (42%) Public Tests
- (42%) Release Tests
- **(8%) Secret Tests**
- (8%) Style

### Which Submission Gets Graded

In this course we grade the highest scoring submission in the submit server after a late penalty has been applied (in case you have a late submission). **The submission selected does not take into account any documentation/style/additional requirements,** therefore you need to complete documentation/style/additional requirements as you complete your work. If you don't, you may get a low score. For example, you could have one submission that has 70 pts of tests points and no documentation, and a second one with 60 pts of tests points and documentation (20 pts). We will grade the first one and not the second (you will get a score of 70 instead of 80) as we select from the submit server the one with the highest tests score.

### Code Distribution

The project's code distribution is available at JavaLoops.zip (https://www.cs.umd.edu/class/fall2021/cmsc131-01XX-03XX/prot/projects/zipFiles/JavaLoops.zip). Download it and import it as you have imported our class examples. The code distribution provides you with the following:

- **programs** - A package (folder) where you will create three programs/classes named **Convert**, **ThrowDie**, and **Access**. In addition, you will find a class called **Support** that provides the method **reverseString**. This method returns the reverse of a string and it can be helpful while implementing this project. You are not required to use this method.
- **tests** - A package (folder) where you will find public tests.
- **expectedResults** - A folder that has the expected results for each of the public test.
- **results** - A folder that the results generated by your code for each of the public tests.

## Specifications

You need to implement three programs: Convert, ThrowDie, and Access. **These programs must be implemented in the programs package otherwise you will not be able to run tests in the submit server.**

Programs

1. **Convert** Program - Write a program that converts a positive integer decimal number (including 0) to octal. To convert a number, use the approach discussed in lab where we keep dividing by the base (in this case 8) using remainders to generate the conversion. **You may not use the Integer.toOctalString method** or any other Java method that generates the conversion for you; if you do, you will lose all the points for this program. You must use loops to compute the octal representation (additional information at the end). The following example illustrates the messages to read data and to display results:

```
Enter decimal number: 10
Octal value: 12
```

2. **ThrowDie** Program - Write a program that simulates throwing a die **n** times. The program will read the number of times to throw the die, a seed and then it will display the throws. Each throw is represented by 9 characters where we use 0 or the period (**.**) character (see sample run below). To generate random values, you need to create a **Random** object with the seed value provided by the user. Use the method nextInt (with 6 as the argument) in order to generate integer random values between 0 and 5; just add 1 to the result to generate the die face number. The following example illustrates the messages to use to read data and display results.

```
How many times to throw a die?: 14
Enter seed: 9
Throw #1
0..
...
..0
Throw #2
0.0
.0.
0.0
Throw #3
0.0
.0.
0.0
Throw #4
0..
...
..0
Throw #5
0.0
0.0
0.0
Throw #6
0.0
...
0.0
Throw #7
0.0
0.0
0.0
Throw #8
...
.0.
...
Throw #9
0..
...
..0
Throw #10
0..
...
..0
Throw #11
0..
.0.
..0
Throw #12
0.0
.0.
0.0
Throw #13
0.0
```

```
.0.
0.0
Throw #14
0.0
.0.
0.0
```

3. **Access** Program - Write a program that reads a string (password) and a number. The program will keep asking the user for a password and a number as long as the user does not provide the expected values ("terps" and 1847, respectively), and as long as the user has not exhausted the maximum number of attempts (3). In addition, the program will stop asking for data if the word "quit" is provided as a password value (in this case, the program will not read the number). The program will print the message "Wrong credentials" when an invalid password or number (or both) is provided. If the user eventually provides the correct credentials, the message "Access Granted" will be displayed; otherwise the message "Access Denied" will be generated. The following example illustrates the messages to use to read data and display results.

```
Enter password: terps
Enter number: 10
Wrong credentials
Enter password: testudo
Enter number: 1847
Wrong credentials
Enter password: terps
Enter number: 1847
Access Granted
```

## Requirements

- Unlike the first project, for this project you need to create the classes (programs). You need to create the classes Access, Convert and ThrowDie in the **programs** package. The classes should appear next to the Support.java class we have left in the **programs** folder.
- You can assume valid data will provided for all the programs. For example, we will not provide a negative number to the convert program.
- You may not use arrays nor ArrayLists. These are data structures that store several elements in a unit.
- **You don't need arrays to generate the die faces; just use output statements that draw the appropiate die face.**
- Verify that your project passes the submit server public and release tests (https://submit.cs.umd.edu/ (https://submit.cs.umd.edu/fall2021)). Your score on the project will be based on results from the submit server and NOT from your Eclipse results.
- **You can only release test your project once you have passed all public tests.** For this project you have 3 tokens in a 22-hour period.
- Do not use System.exit(0) in your programs.
- If the user exhausts all the possible attempts, your program must print

```
Wrong credentials
Access Denied
```

not just "Access Denied".

- If the user enters quit only "Access Denied" should be printed:

```
Enter password: quit
Access Denied
```

- **Do not use nextLine() in Access.java.**
- **If you are failing release tests:**
  - We cannot provide information about release tests.
  - If you are not passing release tests:
    - Make sure you don't have spelling errors in messages a program generates. If you are missing a character or misspelled one, you will not pass a test. Spaces in a line are not an issue; extra blank lines may generate problems.
    - Carefully read the description, making sure your code implements the expected functionality.
    - Read the project clarifications we have posted. Some of them can help you pass tests.
    - Talk to a TA to make sure your understanding of what the code it is supposed to do is correct.
    - There is nothing tricky about release/secret tests; they are based on the information we have provided in the project description and in the clarifications we have posted in Piazza.
- For style we will be focusing on the following items:
  - Good variable names and indentation.
  - Avoid code duplication when possible.
  - Remove any variables you are not using.
  - You do not need to add comments throughout your code.
- A submit server results color guide can be found on the class web page (Resource→Other). A direct link is Submit Server Results Color Guide. (http://www.cs.umd.edu/~nelson/classes/resources/SubmitServerResultsColorGuide.png)
- Make sure you add the 80-character mark in Eclipse so your lines do not exceed 80 characters. Additional information at 80-characters mark (http://www.cs.umd.edu/~nelson/eclipse/other/#80-characters)

**If your code throws an exemption, no results files will be generated. Also, remember to refresh your Eclipse project after running a test so new files can be seen. To refresh a project in Eclipse, right-click on the project and select "Refresh".**

## Converting Between Bases

Below we describe how you convert a number from base 10 to another base. In this example we are converting from base 10 to base 7 **(not necessarily the one used for the project).** We convert by repeated integer division of the base you are going into (integer division means holding onto the remainder). The remainders of the integer division represent the number (in the target base) when they are written in the reverse order. In the following examples, _10 indicates base 10 (e.g., 76_10 means 76 (base 10)), and _7 base 7.

```
76_10 divided by 7 gives you 10 remainder 6
10_10 divided by 7 gives you 1 remainder 3
1_10 divided by 7 gives you 0 remainder 1 so 76_10=136_7

113_10 divided by 5 gives 22 remainder 3
22_10 divided by 5 gives 4 remainder 2
4_10 divided by 5 gives 0 remainder 4 so 113_10 = 423_5
```

## Submission

- Submit your project from Eclipse (within the Java perspective) by right-clicking the project folder and selecting "Submit Project".
- **You need to submit often so you have a backup copy in the submit server.** .
- If you write the project from scratch, without using the provided zip file, you will not be able to submit your work.

## Academic Integrity

Please make sure you read the academic integrity section of the syllabus so you understand what is permissible in our programming projects. We want to remind you that we check your project against other students' projects and any case of academic dishonesty will be referred to the University's Office of Student Conduct (https://www.studentconduct.umd.edu/).

Web Accessibility (https://www.umd.edu/web-accessibility/)