

<p style="text-align: center;">CS 140 Introduction to Computer Science Lab #3</p>

For this lab, students will work individually. If questions arise, try to see if those around you are able to answer them, chances are one of your fellow students has encountered your problem and has figured out how to solve it. Ask the instructor for help as a last resort.

Try your best to complete the following programs today in class. If you cannot complete them all, make sure you complete them prior to the next class. Submit all files together in one .zip file to blackboard.

```
Exam.java  
Pattern.java  
FileIO.java  
PrimeChecker.java  
MultiplicationLearner.java
```

Lab Objectives

- Be able to write Java programs with
 - Decision Structures
 - Input Validation
 - Loops and Nested Loops
 - Sentinel Values
 - File Input and Output
 - Random Numbers
- Be able to test and debug a program

Task #1 Exam

- (1) Rewrite the following program and use a for-loop to replace the while-loop.
- (2) Add statements to verify whether the user input is either 0 or 1. If not, display an appropriate error message and ask the user to enter again until ten correct input values are processed.

```
//analysis of examination results
import java.util.Scanner;

public class Exam
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner(System.in);
        int passes=0, failures=0, students=0, result;
        while (students<10)
        {
            System.out.print("enter result (1=pass, 0=fail): ");
            result = keyboard.nextInt();
            if (result==1)
                passes++;
            else
                failures++;
            students++;
        }
        System.out.println(passes + " passed\n"
                           + failures + " failed");
        if (passes < 5)
            System.out.println("Raise tuition");
    }
}
```

Task #2 Pattern

Write a complete Java program using nested-for loops to produce the following pattern:

```
1 2 3 4 5 6
1 2 3 4 5
1 2 3 4
1 2 3
1 2
1
```

Task #3 File Input and Output

Write a complete Java program that asks the user for the names of three files. The first file should be opened for reading and the other two files should be opened for writing.

The program should read the contents of the first file, change all characters to uppercase, and store the results in the second file at the same time append the results to the third file. At the end, the second file will be a copy of the first file, except that all the characters will be uppercase. If we run the program several times, the third file will hold the contents of all input files that your program has processed. Use any text editor to create several input files that can be used to test the program.

Ex: Suppose `inFile1.dat` contains Monday, Tuesday and `inFile2.dat` contains Wednesday, Thursday, and Friday.

```
tadiatz@fluffy ~ $ java FileIO
enter input filename: inFile1.dat
enter output filename: outFile
enter another output filename (append): appFile
check your output files -- outFile, appFile
```

```
tadiatz@fluffy ~ $ cat outFile
MONDAY
TUESDAY
tadiatz@fluffy ~ $ cat appFile
MONDAY
TUESDAY
```

```
tadiatz@fluffy ~ $ java FileIO
enter input filename: inFile2.dat
enter output filename: outFile
enter another output filename (append): appFile
check your output files -- outFile, appFile
```

```
tadiatz@fluffy ~ $ $ cat outFile
WEDNESDAY
THURSDAY
FRIDAY
tadiatz@fluffy ~ $ cat appFile
MONDAY
TUESDAY
WEDNESDAY
THURSDAY
FRIDAY
```

Task #4 Prime Checker

A prime number is a number that is only evenly divisible by itself and 1. For example, the number 5 is a prime because it can only be evenly divided by 1 and 5. The number 6, however, is not a prime because it can be divided evenly by 1, 2, 3, and 6.

Write a `static boolean` method named `isPrime`, which takes an integer as an argument and returns true if the argument is a prime number, or false otherwise. Demonstrate the method in a complete program.

Task #5 Educational Learning Tools (Challenge)

Computers are playing an increasing role in education. Write a program that will help an elementary school student learn multiplication. Use random number generator to produce two positive one-digit integers. It should then type a question such as:

How much is 6 times 7?

The student then types the answer (if they enter -1 the program will end). Your program checks the student's answer. If it is correct, print one of the following messages randomly and then ask another multiplication question.

Very good!

Excellent!

Nice work!

Keep up the good work!

If the answer is wrong, print one of the following messages and then let the student try the same question again repeatedly until the student finally gets it right. The main purpose for varying the computer's dialogue is to hold the student's attention.

No. Please try again.

Wrong. Try once more.

Don't give up!

No. Keep trying.