Name: Dawson Demien

Student ID: 4001332

Ground rules:

* No person is fair game, besides yourself (and question clarification queries to the instructor, obviously… for the benefit of you pesky would-be lawyers.)
* A generally Google-able (as in, not a person over google chat/nice try) resource is fair game.
* You must do 2 of the challenges. You may do 3 for extra credit (but please do not jeopardize a grade on another exam trying to do this). Please remove the “unused” challenge from your turn in if you only do 2.

First Challenge, Choose one of 3?:

// For one of {Lisp, Clojure, **Scala**} code the following:

// We’re going to help out the Matlab fans out there who think elements are numbered from 1….

// and build a compatibility function for list removal, by….

// removing the X-1 element from a list.

// If there is no X-1 element, handle the error appropriately.  
//  
// Example:  
// removeAt(2, List('a, 'b, 'c, 'd))

// should “actually” remove c. But Matlab would have it that it would be “b”.

// so your should return a c d

// you are free to return the results/outputs in the way easiest for you to manage this challenge

// in terms of data structure or answer style, etc.

**Please cut and paste your code into a document and provide a screen shot of it running.**

import java.io.FileReader

import java.io.FileNotFoundException

import java.io.IOException

object MyClass {

def main(args: Array[String]) {

val nums = Array(5, 66, 78, 33, 89);

removeAt(4, nums)

}

def removeAt (num: Int, arr: Array[Int])

{

var a = num-1

var i = 0

if (num <= 0) {

throw new ArithmeticException("Access denied - The position must be greater than 0");

}

while( i < a)

{

print(arr(i) + ", ")

i += 1

}

var j = num

while( j < arr.length)

{

print(arr(j) + ", ")

j += 1

}

}

}

Good Output:

Text

Description automatically generated

Error Handling:

A screenshot of a computer

Description automatically generated with medium confidence

Second Challenge, Prolog:

// Encode a *Classifier Tool* for something you know.

// Likely places you left your keys (always a problem for me)

// If someone is a member of a particular sort of student org

// (think…. likesPuzzles…. Chess club or Math Club? for example)

// Or, which building someone spends the most time in (major(X) / knowsPerson(Y)).

// plus at least 2 rules that chain the facts (at least 6). Anything pretty much goes here….

// EXCEPT: No Animals. We’ve done that to death.

// Tip - try <https://www.metalevel.at/prolog/expertsystems>

**Please cut and paste your code into a document and provide a screen shot of it running.**

game(overwatch) :- is\_true('is a FPS'), is\_true('Created by Blizzard').

game(apex\_legends) :- is\_true('is a FPS'), is\_true('Created by Respawn').

game(outer\_worlds) :- is\_true('is a FPS'), is\_true('Created by Obsidian').

game(breath\_of\_the\_wild) :- is\_true('is an adventure game'), is\_true('Created by Nintendo').

game(horizon\_zero\_dawn) :- is\_true('is an adventure game'), is\_true('Created by Guerilla Games').

game(sable) :- is\_true('is an adventure game'), is\_true('Created by Shedworks').

game(persona\_5) :- is\_true('is a JRPG'), is\_true('Created by Atlas'), is\_true('Joker is the main character').

game(shin\_megami\_tensei\_5) :- is\_true('is a JRPG'), is\_true('Created by Atlas'), is\_true('The protagonist fuses with a Nahobino').

game(dragon\_quest\_11) :- is\_true('is a JRPG'), is\_true('Created by Square Enix').

is\_true(Q) :-

format("~w?\n", [Q]),

read(yes).

Graphical user interface, text, application

Description automatically generated