**LESSON PLAN**

**CONCEPT #1: DATA TYPES**

Content

* int
* float & double
* bool
* char
* void
* string

Examples

* int + int
* float + float
* int + float
* bool
* char
* string
* string + string
* string + char

**CONCEPT #2: IF STATEMENTS**

Content

* if
* else if
* else
* &&
* ||

Examples

* if (x < 10)
* else if (x < 20)
* else
* if (x < 10 && x > 0)
* else if (x < 20 || x < 0)

**CONCEPT #3: LOOPS**

Content

* for
* while
* do while

Examples

* for
* while
* do while
* inline for

**CONCEPT #3: INPUT**

Content

* cin
* getline

Examples

* cin { ‘hello world’ }
* getline { ‘hello world’ }

**CONCEPT #4: FUNCTIONS**

Content

* Prototypes
* Parameters
* Calls

Examples

* int max(int x, int y)

**CONCEPT #5: ARRAYS**

Content

* Arrays

Examples

* int arr[10]
* arr[0] = x

**CONCEPT #6: OBJECTS**

Content

* struct
* member
* method
* dot operator
* *pointers*

Examples

* object {

int x;

int y;

int add(); }

* object.x
* object.add

**EXERCISE #1: REPORT CARD GENERATOR**

1. Create new console application.
2. Include <string> and <iostream> and add “using namespace std”
3. Declare a string for the student’s name.
4. Declare an array of integers for the student’s grades. A size of 5 should be enough.
5. Gather user input for the student’s name. This can be done using “getline”.
6. Gather user input for each of the elements of your new array. This can be done using “cin” and a “for” loop.
7. Declare an array of chars for the student’s letter grade.
8. Convert the percent value from the integer array and store it in the new char array. This can be done using a “for” loop and series of “if” statements.
9. Output the student’s name and grades. Make it look pretty!

***Going further:***

1. Add prompts for user input, such as “Please enter the student’s name” before trying to gather input.
2. Use functions to get user input and convert percent to letter.
3. Validate user input. Make sure the name doesn’t contain numbers, and make sure the grades don’t contain letters!
4. Create a “Student” object that stores the name, grades, and letters as members and the functions as methods.
5. Write it in as few lines as possible!

**EXERCISE #2: CAESAR CIPHER**

In this project you will use:

1. Input/output
2. Strings

Control Flow:

1. Ask the user to input a secret message.
2. Encode the message using a Caesar cipher. (Ex. A => B, B => C, …, Z => A)
3. Output the encoded message.

Hints:

1. The input will be longer than a single character, likely a sentence or more.
2. Strings are really just arrays of characters, this means you can look at individual characters with “string[0]”
3. One method for doing this is to have a bunch of “if” statements checking each character in the user’s input string and change the letter’s one at a time.

***Going further:***

1. Allow the user to specify the shift of the Caesar cipher. (Ex. a shift of 1 would result in A => B, but a shift of 2 would make A => C)
2. Make a decoder, where the use can input an encoded message and the shift and the program output’s the original message.
3. Write it in as few lines as possible!