Learning Objectives:

1. Write correct programs using pointers to pass parameters to and from subprograms.
2. Write correct programs that dynamically allocate arrays.

To start the lab:

1. watch the Lab 2 Video on the eLearning system
2. download lab2.zip from the eLearning system and unzip/extract it locally on your computer
3. copy the lab2 directory to student.cs.uni.edu using a secure ftp client (winSCP, FileZilla, scp, etc.)
4. log-on to student.cs.uni.edu using Putty/ssh

**Part A:** Using an editor on student.cs.uni.edu open the file displayCircleInfo.c which contains a simple C program to interactively allow the user to enter a circle’s radius and it outputs the circles area and circumference. Notice that the program is split into a main function that acts as the high-level outline for the program and calls three functions: getRadius, calculateAreaAndCircumference, and displayCircleInformation.

Answer the following questions about the displayCircleInfo.c program:

The main calls calculateAreaAndCircumference:

calculateAreaAndCircumference(radius, &area, &circumference);

a) What is the purpose of & symbol in front of the area, and circumference parameters?

**ANSWER** : Says look at the address of area and circumference

b) Why does the radius parameter not have a & symbol in front of it?

**ANSWER:** Because we are already sending in a pointer

The calculateAreaAndCircumference function definition is:

void calculateAreaAndCircumference(double radius, double \* area,

double \* circumference) {

\*area = PI \* pow(radius, 2.0);

\*circumference = 2.0 \* PI \* radius;

} // end calculateAreaAndCircumference

ANSWER: It has been declared as getRadius (function has already received address)

c) What is the purpose of \* symbol in front of the area, and circumference parameters in the function header?

**ANSWER:** Pointer to the double coming from the main

d) What is the purpose of \* symbol in front of the area and circumference in the assignment statements?

**ANSWER:** Follow that address to that memory location

The getRadius function definition is:

double getRadius() {

double radius;

printf("Enter the radius of a circle: ");

scanf("%lf", &radius); // NOTE %lf for double, but %f for float

return radius;

} // end getRadius

e) What is the purpose of double in the double getRadius() { function header?

**ANSWER:** Double is type we declare used to define numeric variables holding numbers with decimal points.

f) What is the purpose of & symbol in front of the radius parameter in the scanf call?

**ANSWER:** scanf needs to know the address of variable to locate

**Part B:** Write a similar program called displayRectangleInfo.c that is split into a main function that acts as the high-level outline for the program and calls three functions:

1. getDimensions - prompts the user and return the length and width of the rectangle,
2. calculateAreaAndCircumference - calculates and returns rectangle’s area and circumference, and
3. displayRectangleInformation - displays the rectangle’s information to the user in a nicely formatted fashion

Write, compile, debug your program on student.cs.uni.edu. Use the script command to capture the user interaction when you run the final version.

**Part C:**  Using an editor on student.cs.uni.edu open the file averageScores.c which contains a simple C program to interactively allow the user to enter a collection of scores to be averaged.

Answer the following questions about the averageScores.c program:

a) What is the maximum number of scores that can be handled by this program?

**ANSWER:** 100

The main calls getScores as:

getScores(&numberOfScores, scores);

b) What is the purpose of & symbol in front of the numberOfScores parameter?

**ANSWER:** It is address of operator and looks up the address of numberofscores

c) Why does the scores parameter not need a & symbol in front it?

**ANSWER:** The pointer is already set on scores array when the new array is made

The getScores function definition is:

void getScores(int \* count, double scores[]) {

double score;

printf("Enter scores one at a time (enter -1 to quit.)\n");

\*count = 0;

while (1) { // infinite loop any nonzero integer is True

printf("Enter a score (or -1 when done): ");

scanf("%lf", &score);

if (score < 0.0) {

break;

} // end if

scores[\*count] = score;

(\*count)++;

} // end while

} // end getScores

d) Why are the parenthesis necessary when incrementing the count ? (\*count)++;

**ANSWER:** They are needed because of the order of precedence. We need to add the values first in the array before changing old value with new.

**Part D:**  Using an editor on student.cs.uni.edu open the file averageScores2.c which also interactively averages a collection of scores, but this program:

1. asks the user to enter the number of scores first, then
2. dynamically allocates an array just big enough to hold the scores using malloc (memory allocate)

Recall that malloc takes as a parameter the **size of the array in bytes** and return a pointer to the first element of the dynamically allocated array in the heap. Typically, you use the sizeof function to determine the size of a single element and multiply by the number of elements to calculate the size of the array in bytes. The malloc function returns “generic” void \* pointer type which must be cast to a pointer of the appropriate type.

Answer the following questions about the averageScores2.c program:

a) When the main program starts execution where does the scores pointer point?

**ANSWER:** It points at numberOfScores because scores array is the number entered by user then getScores knows what size of array to make.

The main calls getScores as:

getScores(&numberOfScores, &scores);

b) What is the purpose of & symbol in front of the scores parameter?

Look at the address if scores

**ANSWER:** Address of operator and points at scores. When main executed, the pointer is at numberOfScores

The getScores function definition is:

void getScores(int \* count, double \*\* scores) {

double score;

double \* localScoresPtr;

int i;

printf("Enter the # of scores you will be entering: ");

scanf("%d", count);

localScoresPtr = (double \*) malloc(sizeof(double)\*(\*count));

for (i = 0; i < \*count; i++) {

printf("Enter a score: ");

scanf("%lf", &score);

localScoresPtr[i] = score;

} // end for

\*scores = localScoresPtr;

} // end getScores

c) What is the purpose of \*\* symbols in front of the scores parameter in the function header?

**ANSWER:** This is a pointer to a pointer when you want to change the value of a function through a pointer

d) Explain all parts of the assignment statement:

localScoresPtr = (double \*) malloc(sizeof(double)\*(\*count));

**ANSWER:**

* localScoresPtr creates an unknown array size
* double \* points at the first value of array
* malloc(sizeof(double) dynamically allocates array by getting the length of array from count
* count tells size of array

e) What type of value is assignment by the assignment statement? \*scores = localScoresPtr;

**ANSWER:** localScoresPtr is the araay created by malloc and the values are stored in \*scores.

**Part E:** Complete the program called displayMultiplicationTable.c that asks the user to enter two integers: value1 and value2, then prints a multiplication table with:

1. rows labeled 1 to value1,
2. columns labeled 1 to value2, and
3. each value in the table being the product of the corresponding row and column label

The main function acts as a high-level outline for the program and calls three functions:

1. getValues - allows the user to enter and returns value1 and value2 values
2. calculateRowProducts - passed a row value, the value2, and returns an array of products from (row x 1) to

(row x value2)

1. printTableHeading - passed the value1 and value2 values and prints the table header
2. printRow - passed a row value, the value2, and corresponding rowProducts array which is printed

**Submit lab2.zip containing question answers and completed programs on eLearning system**