Lab 9

Submission Instructions:

1. Create a folder named Lab9_LastName_FirstInitial (e.g. Lab9_Neal_T).

2. In your folder, place a **PDF** file containing your answers to questions with a ♢.

3. Copy your directories containing your programs for questions with a ♠ into the folder; **these directories should only contain files needed to run your program, which may include one or more of the following file types: .cpp, .h., and .txt**. Do NOT include the full project (e.g., solution file). Test your program on CIRCE before submitting by compiling and running with g++. Your file containing main() should **always** be named main.cpp.

4. Ensure that all programs have block comments at the very beginning (starting at the first line) in the file containing main() with your name and the program's description. **The block comment's format should be identical to what's provided in Figure 2-1.**

5. Use single-line comments to describe your code's functionality as needed.

6. Do not submit anything for questions with a ♣.

7. Zip the folder and submit it via Canvas.

♢ = 5 points each, ♠ = 15 points each

---

1. ♣ Read *Chapter 17: How to work with memory and pointers.*

2. ♢ To get the value of the object that a pointer points to, you use
   a. the indirection operator
   b. the address of operator
   c. a self-reference
   d. the this pointer

3. ♢ A pointer can be referred to as a compound type because
   a. it is an object that points to the memory address of another object
   b. it can be dereferenced to get the value of the object it points to
   c. it can be used to access the public data members of the object it points to
   d. it stores both a memory address and a data type

4. ♢ Given two double variables named wages and salary, which of the following statements could you use to define pointers to both variables in a single statement?

```
a. int* ptr1 = &wages, ptr2 = &salary;
b. int *ptr1 = &wages, *ptr2 = &salary;
c. int* ptr1 = wages&, ptr2 = salary&;
d. int *ptr1 = wages&, *ptr2 = salary&;
```

5. ♢ Which of the following statements apply to pointer variables but not reference variables?
   a. They can be null.
   b. They're objects with their own memory addresses.
   c. You need to use the indirection operator to access their underlying values.
   d. All of the above
   e. b and c only

6. ◇ What does the statement that follows do?

```
int* age = new int(49);
```

a. It defines an integer with the value 49 and stores it in the variable named age.
b. It allocates free store memory for an integer with the value 49 and stores the value in the variable named age.
c. It allocates free store memory for an integer with the value 49 and stores a pointer to that memory in the variable named age.
d. It allocates 49 bytes of free store memory and stores a pointer to that memory in the variable named age.

7. ♠ **MatheMagics**
Create a program that provides the user with options for performing mathematical operations and return the result of each operation using pointers. Save your file(s) in folder lab10_q7.

Console

```
The MatheMagics App

Command Menu
-------------
square: perform squaring of a number
cube:   perform cubing of a number
pow:    raise a number to an integer power
quit:   quit the program.

Command: square
Enter a number to square: 4
Result: 16

Command Menu
-------------
square: perform squaring of a number
cube:   perform cubing of a number
pow:    raise a number to an integer power
quit:   quit the program.

Command:  pow
Enter the base: 2
Enter the integer exponent: 3
Result: 8

Command Menu
-------------
square: perform squaring of a number
cube:   perform cubing of a number
pow:    raise a number to an integer power
quit:   quit the program.

Command: quit
Bye!
```

Specifications

- Write three mathematical functions named square(), cube(), and pow() without using any of the built-in math libraries such as the cmath library.
- For the square() and cube() functions, accept the argument for the number to be squared or cubed as a pointer to a double and store the result in that same variable.
- For the pow() function, accept the arguments for the base and exponent by value, but accept the argument for the result as a pointer to a double and store the result in that variable.
- By definition, any base to the power of 0 is equal to 1.

8. ♠ **Dynamic Pets**

   Create a program that adds the Pet objects specified by the user to a vector of Pet objects and use dynamic (free store) memory to store each Pet object. Save your file(s) in folder lab10_q8.

   Console

```
Dynamic Pets

Enter the pet’s name:  Rover
Enter the pet’s species:  dog
Add another pet? (y/n):  y

Enter the pet’s name:  Felix
Enter the pet’s species:  cat
Add another pet? (y/n):  y

Enter the pet’s name:  Tweeterz
Enter the pet’s species:  bird
Add another pet? (y/n):  n

Rover is a(n) dog
Felix is a(n) cat
Tweeterz is a(n) bird

Bye!
```

   Specifications

   - Create a class named Pet whose full definition is stored in a header file named Pet.h.
   - The Pet class should have private data members to store the name and species of the pet. It should have a constructor that accepts the name and species and sets the private data members. And it should have getter methods for the private data members.
   - Dynamically allocate memory for each Pet object at runtime.
   - Use a vector to store pointers to each Pet object.
   - Make sure to clean up (deallocate) the memory before the program exits.

9. ♠ Leave any questions that you have concerning the content presented in Chapter 17 in this week’s Q&A (Week 11) discussion and/or provide answers or feedback for other students. If you have none, provide other resources relevant to this week’s topics that you found helpful or feel may help others.