Object-oriented Programming in C++

Practical Worksheet 5

Questions

*(Questions marked * need to be submitted to the NOW dropbox.)*

(Make a Visual Studio solution file by creating a new Visual Studio project for the first question. Add .cpp files, etc. Add further Visual Studio projects to this solution for each question.)

LECTURE 5

30. Modify your class from question 29, so that the make and owner are now implemented using a pointer to `char`. You now need to:

    - allocate memory in the constructor for make and owner (using `new`);
    - add a destructor function and code to show it is called (`cout << "Destructor called";`);
    - modify the other functions where necessary (e.g. `changeOwner()`).

    Do you need to alter the main function?

31*. Modify your program from question 30 to create a second Car object as a copy of the first one (use '=' to assign it), then change the owner (by calling `changeOwner()`) and display both cars again. Do you notice any errors?

33*. Rewrite function `computeCircle` from question 19 (sheet 1) using pointers, not reference parameters:

- The function should have two parameters of type `float*` that are pointers to area and circumference (e.g. `void computeCircle(float* a, float* c, float r)`).
- The function call must pass the addresses of the variables (e.g. `computeCircle(&area, &circumference, radius`).

The new function demonstrates how with the C++ language more than one value can be passed back to the calling function (without using an array, structure or class).

34. Write a program which dynamically allocates memory to store a person's name (one or more names separated by spaces) typed in by the user and then displays it on the screen.

    - Declare a character pointer name (of type `char*`) and a string (array of `char`) of size 30, say.
    - Store the name initially in the string before allocating the required memory

      (e.g., `personPtr = new char[size];`).
    - You can use the string function `strlen()` to find out how many characters are in the string.

35. Modify the program from question 34 by declaring an array of `char`*, called `persArray`, so that 10 names can be stored in dynamically allocated memory and then displayed. Allow the user to type in the names on separate lines, terminating with an empty line.

    - Each name should be stored temporarily in the string before the required memory is allocated.
    - The names can be displayed using `cout`, e.g. for the first one: `cout << persArray[0];`.

36. Modify the code from question 35 by writing a function `displayNames` to display all the names. Pass the array of names to the function as a pointer of type `char**`. What else do you need to pass?

37*. Now modify the program from question 36 to allow the user to enter the total number of names initially and then allocate the array of pointers dynamically (using a pointer of type `char**`).
