

1.

Erzeugen Sie ein `stash`-Objekt, das `doubles` enthält. Fügen Sie dem Objekt 25 `doubles` hinzu und geben Sie die `doubles` auf der Konsole `cout` aus.

2.

Erstellen Sie eine Funktion mit einem Parameter vom Typ `char *`. Allokieren Sie mit `new` dynamisch Speicherplatz für ein `char`-Array der Grösse des via Pointer übergebenen `char`-Array. Kopieren Sie dann mit Array-Indexierung alle `chars` vom Argument in das dynamisch allozierte Array und geben Sie den Pointer der Kopie aus der Funktion zurück.

Testen Sie in `main()` Ihre Funktion, indem Sie der Funktion ein String-Literal übergeben. Das Resultat der Funktion ziehen Sie heran, um es der Funktion neuerlich zu übergeben. Geben Sie schliesslich die `char`-Arrays sowie die Start-Adressen an `cout` aus, um zu zeigen, dass die `char`-Arrays an unterschiedlichen Adressen abgelegt sind. Verwenden Sie am Schluss `delete` auf korrekte Weise.

3.

Kreieren Sie einen `stack`, der `stashes` enthält (vgl. Folien und Buch). Jeder `stash` enthalte fünf Zeilen einer Input-Datei. Kreieren Sie `stashes` mit `new`. Lesen Sie jetzt eine Datei (via `stashes`) in den `stack` ein. Geben Sie darauf die Datei Zeile für Zeile in umgekehrter Reihenfolge an `cout` aus.

4.

Kreieren Sie eine `struct T`, die eine `int` und einen Pointer auf eine nächste `struct T` enthält. Schreiben Sie eine Funktion innerhalb von `T`, die eine `int size` entgegennimmt. `size` repräsentiere die Länge einer zu bildenden Linked List, die in der Funktion erstellt werde. Der Kopf der Liste sei selbst eine Instanz des Typs `struct T`, die mit in `size` berücksichtigt sei. Das letzte Element der Liste enthalte einen `NULL`-Pointer.

Schreiben Sie mindestens eine weitere Funktion, die durch die Liste iteriert und jeweils die `int` und den Pointerwert auf einer Zeile an `cout` ausgibt. Vergessen Sie das cleanup nicht.

5.

Kreieren Sie eine Klasse und einen globalen `friend`, der die privaten Daten der Klasse manipuliert. Stellen Sie das Vorher und Nachher an `cout` dar.

6.

Kreieren Sie eine Klasse `Hen`. In dieser Klasse kreieren Sie eine Klasse `Nest` und in dieser wiederum eine Klasse `Egg`. Jede Klasse weise eine Methode `display()` auf. Erstellen Sie in `main()` eine Instanz jeder Klasse und senden Sie `display()` an jede Instanz.

7.

Fakultativ (erfordert Kenntnisse zu `make`):

Modifizieren Sie `Cheshire` in `Handle.cpp` (vgl. Folien und Buch) und verifizieren Sie mittels Verwendung von `make`, dass nur dieses eine File rekompiliert und gelinkt wird. Verifizieren Sie insbesondere, dass `UseHandle.cpp` nicht neu kompiliert wird.

Variante: Untersuchen Sie, wie sich Ihre IDE diesbezüglich verhält.