Übung 10 infT

1.
Weisen Sie in einem entsprechenden Beispiel die Richtigkeit der folgenden Aussage nach:
"Returning a value as a **`const`**ant means the returned value can't be modified." (CppIntro1V16.pdf, Folie 166).

2.
Auf Folie 169 ist Folgendes festgehalten:
"If a function returns a class object as a **`const`** the **`return`** value of that function can't be an lvalue. (lvalue: it can't be assigned to or otherwise modified)."

3.
Testen Sie den Code der Datei *constReturnValues.cpp* (Folien 170, 171). Welche Meldungen liefert Ihr C++-Compiler?

4.
Beim Übergeben eines Objekts in eine Funktion wird das Objekt kopiert. Zeigen Sie, dass dem so ist.

5.
Beim Zurückgeben eines Objekts aus einer Funktion wird das Objekt ebenfalls kopiert. Zeigen Sie auch dies an einem geeigneten Beispiel.

6.
Experimentieren Sie mit temporären Objekten (vgl. Folien 171, 172). Welche Warnung oder Fehlermeldung gibt Ihr Compiler aus?

7.
Eckel behauptet (Folien 174, 175):

```
// constPointer.cpp -- Constant pointer arg/return
const int * const w() {
  static int i;
  return &i;
}

int main() {
  const int * const cip2 = w();
  const int *cip3 = w();          // ???
  return 0;
}
```

Ist die Zeile `// ???` korrekt? Begründung?

8.
Entwerfen Sie eine Klasse mit zwei Memberfunktionen. Die Funktionen sollen dieselben Köpfe aufweisen, ausser dass die eine Funktion als konstante Memberfunktion gekennzeichnet sei. Zeigen Sie jetzt, dass der Compiler eigenständig die jeweils richtige Methode verwendet, wenn Sie mit einem konstanten bzw. nicht-konstanten Objekt arbeiten.

9.
Welche Fehlermeldungen liefert das folgende Programm?

```
/* constTypeQualifier.cpp */
#include <iostream>
using namespace std;

void tryToModifyArray(const int b[]) {
  b[0] /= 2;
  b[1] /= 2;
  b[2] /= 2;
}


int main() {
  int a[] = {10, 20, 30};
  tryToModifyArray(a);
  cout << a[0] << ' ' << a[1] << ' ' << a[2] << endl;
  return 0;
}
```

10.
Das folgende Programm liefert eine Fehlermeldung. Welche Abhilfe/n schlagen Sie vor?

```
/* constMemberArray.cpp */
#include <iostream>
using namespace std;

class Test {
  const int arr[3];
public:
  Test(int, int, int);
  void printElems() const;
};

Test::Test(int i, int j, int k) : arr[0](i), arr[1](j), arr[2](k) { }

void Test::printElems() const {
  cout << "Test object array: arr[0] : " << arr[0]
       << ", arr[1] : " << arr[1]
       << ", arr[2] : " << arr[2] << endl;
}


int main() {
  Test t1(0, 1, 2), t2(-3, 5, -73);
  t1.printElems();
  t2.printElems();
  return 0;
}
```

11.
Testen Sie das `Quoter`-Beispiel (Folien 192 - 194). Untersuchen Sie die Lösung dann in den Details.