4.4.A Von C nach C++/Referenzen – Arbeitsblatt

Aufgabe 1: Setzen einer Variablen in einer Funktion I

Im folgenden Programm soll die Variable `y` mit Hilfe der Funktion `setzenull` auf den Wert `0.0` gesetzt werden.

```
1  int main(void)
2  {
3      double y=3;
4      cout << y << endl;
5      setze_auf_null(y);
6      cout << y << endl;
7      return 0;
8  }
```

a) Warum ist die Lösung der Aufgabe in C mit diesem Funktionsaufruf nicht möglich?

b) Wie müsste der Funktionsaufruf in C aussehen, damit die Aufgabe lösbar ist?

c) Erstellen Sie die Funktion `setzenull`, die auf C++-Art die Aufgabe durch Benutzung einer Referenz löst!

Aufgabe 2: Setzen einer Variablen in einer Funktion II

Im folgenden Programm werden zwei Variablennamen an die Funktion `makemax` übergeben. Die Funktion soll die kleinere der beiden auf den Wert der größeren Variablen setzen.

```
1  int main(void)
2  {
3     int a, b;
4     cout << "Bitte␣a␣eingeben:"; cin >> a;
5     cout << "Bitte␣b␣eingeben:"; cin >> b;
6     cout << "rufe␣makemax(a,b)␣auf..." << endl;
7     makemax(a, b);
8     cout << "Wert␣fuer␣a:" << a << endl;
9     cout << "Wert␣fuer␣b:" << b << endl;
10    return 0;
11 }
```

a) Warum ist die Lösung der Aufgabe in C mit diesem Funktionsaufruf nicht möglich?

b) Wie müsste der Funktionsaufruf in C aussehen, damit die Aufgabe lösbar ist?

c) Erstellen Sie die Funktion `makemax`, die auf C++-Art die Aufgabe durch Benutzung von Referenzen löst!

Aufgabe 3: Vertauschen der Inhalte zweier Variablen

Die Funktion `swapit(a,b)` soll die Inhalte der beiden `int`-Variablen `a` und `b` vertauschen.

a) Warum ist die Lösung der Aufgabe in C mit diesem Funktionsaufruf nicht möglich?

b) Erstellen Sie die Funktion `swapit`, die auf C++-Art die Aufgabe durch Benutzung von Referenzen löst!

c) Binden Sie die Funktion in ein Programm ein, mit dem Sie zeigen können, dass Ihre Funktion richtig arbeitet!

Aufgabe 4: Ersatz für die Funktion `scanf`

Sie möchten nach `printf` jetzt auch `scanf` durch eine Funktion ersetzen, die keinen Platzhalter braucht und trotzdem für Variablen verschiedener Datentypen funktioniert. Außerdem möchten Sie, dass man nicht mehr daran denken muss, vor dem Parameter ein `&`-Zeichen zu schreiben.

Die Funktion `read` soll beide Wünsche erfüllen: Die Funktionsaufruf `read(x)` liest einen Zahlenwert von der Tastatur und schreibt ihn in die Variable `x`. `x` darf vom Typ `int` oder vom Typ `double` sein.

a) Welche C++-Eigenschaft benutzen Sie, damit `read` für mehrere Datentypen funktioniert?

b) Welche C++-Eigenschaft benutzen Sie, damit vor dem Parameter `x` kein `&`-Zeichen stehen muss?

c) Schreiben Sie `read`; testen Sie Ihr Ergebnis durch Aufrufe mit verschiedenen Datentypen!