Arbeitsblatt 2

# Aufgabe 1

* Ein lvalue muss auf einen lvalue und nicht auf einen rvalue zeigen
* Die variable Zahl 42 ist ein rvalue 🡪 Funktioniert nicht
* Die konstante Zahl ist ein lvalue 🡪 Funktioniert

# Aufgabe 2

|  |
| --- |
| #include <iostream>  using namespace std;  void doPointers() {  int i = 'a'; // Variabler Wert  const int ic = i; // Konstanter Wert  const int \*pic = &ic; // Pointer auf konstanter Wert  //int \* const cpi = &ic; // Konstanter Pointer auf variabler [aber eigentlich konstanter] Wert --> Fehler  const int \*const cpic = &ic; // Konstanter Pointer auf konstanter Wert  }  void doReferences() {  //int &i = 'a'; // Referenz auf variabler Wert --> Fehler  int i = 'a';  const int ic = i; // Konstanter Wert  //const int &ric = &ic; // Referenz auf Konstante aber mit Dereferenzierung (Pointer wird zurückgegeben) --> Fehler  //int &const rpi = &ic; // Konstante Referenz auf variabler [aber eigentlich konstanter] Wert und mit Dereferenzierung (Pointer wird zurückgegeben) --> Fehler  //const int &const cpic = &ic; // Konstante Referenz auf konstanzer Wert aber mit Dereferenzierung (Pointer wird zurückgegeben) --> Fehler  }  int main(int argc, char \*argv[]) {  doPointers();  doReferences();  return 0;  } |

# Aufgabe 3

|  |
| --- |
| #include <iostream>  using namespace std;  void swapByPointer(int \*a, int \*b) {  int tmp = \*a;  \*a = \*b;  \*b = tmp;  }  void swapByReference(int &a, int &b) {  int tmp = a;  a = b;  b = tmp;  }  int main(int argc, char \*argv[]) {  int a = 5, b = 10;  swapByPointer(&a, &b);  cout << a << " " << b << endl;  a = 5, b = 10;  swapByReference(a, b);  cout << a << " " << b << endl;  return 0;  } |

# Aufgabe 4

Sehr wahrscheinlich wird eine Segmentation Fault geworfen, da die Variable nach dem Verlassen der Methode out of Scope gegangen ist.

# Aufgabe 5

|  |
| --- |
| #include <iostream>  #include <cmath>  using namespace std;  void sinus(int a, double b) {  double bb = b / 6.28;  for (int y = a; y >= -a; y--) {  for (int x = 0; x < b; x++) {  cout << ((int) round(a \* sin(x / bb)) == y ? '\*' : ' ');  }  cout << endl;  }  }  int main(int argc, char \*argv[]) {  sinus(50, 100);  return 0;  } |