

- Herausforderungen beim Programmieren in C *durch Programmierer nicht definiert*
 - In vielen Fällen undefiniertes Verhalten möglich (z.B. nicht initialisierter Speicher)
 - Nur eingeschränkte Portabilität (Teile der Programmsemantik können vom Compiler und/oder Ausführungsplattform abhängig sein) *(z.B. Big-/Little Endian)*
 - Wenig ausdrucksstark, keine Abstraktionen wie Listen, Tupel, Mengen, usw.
 - An vielen Stellen Verzicht auf Sicherheit, der Programmierer kann die Abstraktionen der Programmiersprache (Variablen, Objekte, Kontrollstrukturen, usw.) relativ leicht verlassen
- **Grundannahme: Programmierer weiß, was er tut**
 - Bedeutet auch: Viele Programmfehler können nicht zur Übersetzungszeit erkannt werden, fehlerhafte Programme werden häufig vom Compiler akzeptiert
 - Programmieren in C erfordert ein hohes Maß an **Disziplin und Präzision**
 - Kompilierbarkeit eines Programm beweist nicht viel
 - Fehler in C-Programmen können zur Laufzeit scheinbar unerklärliches oder zufälliges Verhalten verursachen und sehr schwer zu finden sein *(Heimrechner kann andere Ergebnisse liefern als Rechner in der FH)*