

Allgemeines / Grundlagen (aus Klasse 10, Python)

- Computeraufbau, Informatikbegriff, geschichtliche Wurzeln der Informatik, Teilgebiete
- Grundbegriffe (Daten, Algorithmen, Codierung, EVA-Prinzip)
- Codierung von Daten (Text > ASCII-Code, Sound > Sampling, Bilder > Pixel- und Vektorgrafiken, Zahlen > Binärcode)
- Algorithmenentwurf und deren Darstellungen (Flussdiagramm, Struktogramm, Schreibtischtest, Variablenkonzept)
- Einführung in Python (Installation, Nutzung, Ein- und Ausgaben, Variablen, Datentypen, Kontrollstrukturen, Listen- und Stringoperationen, Casting mit Fehlerbehandlung)

Algorithmen / Imperatives Programmieren (Python)

- Algorithmenentwurf und -darstellung (Flussdiagramm, Struktogramm, Schreibtischtest, Quelltext)
- Sprachübersetzer - Compiler und Interpreter, Fehlersuche - Debugger, Fehlerarten beim Programmieren - Syntaxfehler, Laufzeitfehler, inhaltlich logische Fehler
- Prozedurales Programmieren mit Python - Grundlagen siehe oben, Zerlegungsprinzip: Top-Down-Methode, Funktionen, Parameter, Rückgaben, globale und lokale Variablen - Seiteneffekte

Sprachen, Grammatiken und Automaten (AtoCC - Grammatik- und Automateneditor)

- Natürliche und formale Sprachen - Unterschiede, Probleme
- Einteilungen von Programmiersprachen (nach Generationen, Sprachtypen)
- Grammatik einer formalen Sprache - $G=(N,T,P,s)$, Entwurf und Analyse von Grammatiken, Darstellung von Produktionsregeln mittels (E)BNF und Syntaxdiagrammen, Ableitungsbaum
- Automaten als Modelle - Arten von Automaten, Zusammenhang erkennende Automaten - Sprachtypen nach Chomsky (informativ)
- Endliche Automaten (DEA und NEA), Entwurf und Analyse endlicher Automaten, Darstellung von Automaten (Zustandsdiagramme, Übergangstabelle, Funktionsschreibweise)
- Zusammenhang Grammatiken/Automaten und Sprachübersetzer (Compiler/Interpreter), einfache Automaten (Syntaxprüfungen) mit Python prozedural programmieren

Datenbanken / Deklaratives Programmieren (MySQL)

- Einführung in Datenbanken - Aufbau, Aufgaben, Arten von DBMS, Vorteile und Probleme, Beispiele in der Realwelt
- Entwurf relationaler Datenbanken - Miniwelt, Modell der Miniwelt, ER-Diagramm, Relationenschema, Umsetzung von Beziehungen (Beziehungstypen m:n, 1:n, 1:1, min-max-Notation beachten) im Relationenschema - Grundregel, Optimierungen, Einsatz von Fremdschlüsseln

- Realisierung relationaler Datenbanken mit MySQL auf Konsolenebene (Definition, Eingabe von Daten), Optimierung der Umsetzung von Beziehungen
- einfache Abfragen - Projektion, Selektion und tabellenübergreifende Abfragen unter Nutzung der Primär-Fremdschlüssel-Beziehungen, ordnen, gruppieren, Aggregatfunktionen etc. nutzen
- Entwurf und Realisierung komplexer Datenbanken incl. tabellenübergreifender Abfragen
- Benutzerrechte, Daten ändern usw. (nur informativ)

Rechnerarchitektur, Assembler und Schaltungsentwurf (VNR-Simulator, Crocodile Physics)

- Grundlagen der Rechnerarchitektur - Datencodierung (Wiederholungen: ASCII / Grafiken / Binärcode, dann Zweierkomplement, Gleitpunktdarstellung)
- Von-Neumann-Modell und Von-Neumann-Zyklus (Grundsätze, Aufbau, Funktionsweise)
- Unterschiede heutiger Rechner zum Von-Neumann-Modell (Cache, Pipelining, Parallelität durch Mehrkernprozessoren, technische Erweiterungen und Verbesserungen)

- Programmierung eines Von-Neumann-Rechners (Simulator) in Maschinencode bzw. Assembler (nur direkte Adressierung), Module zur Ein- und Ausgabe von Zahlen nutzen

- Technische Rechnerarchitektur - Grundlagen, Halbleiter, Dotierungsarten, Dioden, bipolare Transistoren, Verwendung von Transistoren in logischen Grundschaltungen (Gatter)

