

Modulbezeichnung (Kürzel)	Programmierung II (PR2)
Modulbezeichnung (eng.)	Programming II
Semester (Häufigkeit)	2 (jedes Sommersemester)
ECTS-Punkte (Dauer)	5 (1 Semester)
Art	Pflichtmodul
Studentische Arbeitsbelastung	15 h Kontaktzeit + 135 h Selbststudium
Voraussetzungen (laut BPO)	
Empf. Voraussetzungen	Programmierung I
Verwendbarkeit	BORE
Prüfungsart und -dauer	Klausur 2 h oder mündliche Prüfung
Lehr- und Lernmethoden	Multimedial aufbereitetes Online-Studienmodul zum Selbststudium mit zeitlich parallel laufender Online-Betreuung und regelmäßigen virtuellen Lehrveranstaltungen
Modulverantwortliche(r)	C. Link
<p>Voraussetzungen für die Vergabe von Leistungspunkten</p> <p>Studienleistung (Programmierlabor): Erfolgreiche Bearbeitung von 3 Einsendeaufgaben (1 CP). Bewertet mit 'Bestanden'</p> <p>Prüfungsleistung (4 CP): Bestehen der Prüfung (Klausur oder mündliche Prüfung)</p>	
<p>Qualifikationsziele</p> <p>Die Studierenden demonstrieren, dass sie über ein mentales Modell von Informationsdarstellung und Programmablauf verfügen, indem sie einen Rechner (mit rekursiv absteigendem Parser) für Bool'sche Ausdrücke konstruieren. Die Studierenden können</p> <ul style="list-style-type: none"> • mit den Besonderheiten des Programmablaufs bei der Mikrocontrollerprogrammierung umgehen, indem sie in einem Programm asynchrone Ereignisse mit den adäquaten Sprachmitteln behandeln und die geeigneten Datenstrukturen auswählen. • Klassen entwerfen, um mit angepassten Typen Aufgabenstellungen besser (abstrakter) umsetzen zu können, indem sie eine oder mehrere Klassen deklarieren und definieren, welche in einem vorgegebenen Programm verwendet werden. • mittels der wesentlichen Konzepte der objektorientierten Programmierung Programme entwerfen, indem sie diese Konzepte bei der Implementierung einer Aufgabenstellung verwenden. • die verschiedenen Arten des Polymorphismus differenzieren, um mit dem jeweils passenden Code von konkreten Typen zu abstrahieren. • ein vorgegebenes Programm mit den passenden Arten polymorpher C++-Sprachmittel ausstatten und bezüglich der Lesbarkeit und Wartbarkeit verbessern. • wesentliche Teile der C++-Standardbibliothek anwenden; hierbei können sie für kleine Problemstellungen die richtigen Bibliotheksteile benennen. • ausgewählte Bibliotheksteilen bei der Implementierung einer Aufgabenstellung benutzen und einschätzen, welche die dazu besser geeigneten sind. • (einige wenige) Idiome und Patterns auf eigenen Code adaptieren. • eine gegebene Problemstellung analysieren, in kleinere Teile zerlegen und mit den jeweils angemessenen C/C++-Sprachmitteln implementieren. 	

Lehrinhalte

Berechnungen auf wichtigen Datentypen Elementare Datentypen, char-ASCII-Glyph, I/O, Bitschieberei; call stack, Rekursion; free store

Nichtlinearer Programmablauf Reentranz; event-based programming vs. thread-based; interrupt service routines; locking, lock-free data structures

Benutzerdefinierte Datentypen Komposition neuer benutzerdefinierter Datentypen, um Ausdrucksmächtigkeit zu erhöhen; operator overloading; einfache Klasse ohne Vererbung; Typumwandlung (implizit/explicit)

Objektorientierte Programmierung Information hiding; subtyping; Interface vs. Implementation; aggregation vs. composition; Identität vs. Äquivalenz; Delphi-style OO (i.e. only free store objects, no assignment operator, only explicit ctors)

Polymorphismus Subtyping polymorphism mit virtuellen Methoden; Einordnung des bekannten Polymorphismustypen ad-hoc polymorphism; Generic Polymorphism (templates)

C++-Standardbibliothek Std Container; Std Algorithmen; moderner C++-Programmierstil

Idiome und Muster Constructional: virtual ctor, factory, ...; Resource Acquisition Is Initialization (RAII); Design-by-contract (DBC)

Eleganz und diesbezügliche Hindernisse Klassendesign; Schwierigkeiten bei der Kombination von Sprach-Features; Objekte und Pointer (copy ctor, operator=, ...); Vererbung und operator overloading; Vererbung und container

Literatur

Stroustrup, Bjarne (2014): The C++ programming language. [C++ 11]. 4. ed., 2. print. Upper Saddle River, NJ: Addison-Wesley.

Kirch, Ulla; Prinz, Peter (2013): C++ - das Übungsbuch. 4., überarb. Aufl. Heidelberg, München, Landsberg, Frechen, Hamburg: mitp.

Lehrveranstaltungen

Dozenten/-innen	Titel der Lehrveranstaltung
C. Link	Programmierung II
C. Link	Praktikum Programmierung II