

PRÜFUNGSLEISTUNG im

STUDIENGANG

Software Technology

MODUL: CPL FACH: Concepts of Programming Languages

DATUM: 02.07.2021 NAME:

ZEIT: 08:30-10:30 SEMESTER: ST 1

PRÜFER: Prof. Dr. Peter Heusch

MODUL-TEILLEISTUNGEN TEILPUNKTE GESAMTPUNKTE NOTE

FACH 1: CPL

HILFSMITTEL: Original Slides + Notes + Seminar Papers

ANLAGEN:

Please read the following general rules before starting the exam:

- Put your name on every sheet that you use for noting your exam, but do not put your name on otherwise empty sheets
- Do not write final answers on concept paper, do not write final answers with pencils, do not use your own paper
- Write legible, leave at least 5 mm of empty space between your lines and at least 40 mm of empty space at the right margin
- **Put the solutions for questions 5 and 6 on a separate sheet of paper. Failing to do so will result in 50% deduction of points for these two questions.**
- The points for every question reflect the amount of minutes you need to answer it.

1. Question (30P)

Answer the following questions concerning the languages from the seminar. Every question has 3 points. Some questions contain a limitation of results after the question. Failing to comply (and giving more names) results in fewer points! Also denote in one sentence why the language of your choice belongs into that list.

- 1) What is the central difference between macro languages and other languages? Which languages in the seminary were macro languages
- 2) Do you think that assembler language plays a role in web programming? If yes, give an example, if no give arguments why not.
- 3) Which languages would you use to solve problems of artificial intelligence? Give three names.
- 4) Both Python and FORTH can be used interactively. However, Python allows to call a function recursively while it is being defined while FORTH needs the word recurse. What does this tell about the languages?
- 5) Which language of all the languages in the seminar has the easiest scanner?

- 6) Describe typical features of interpreted and compiled languages. Give two names for every category.
- 7) Which languages are specialized on web environments? Give three names.
- 8) Which languages were designed for commercial programs? Give three names.
- 9) Recently a new function named „lambda“ was added to Excel that allows the creation of user defined functions (with arrays as parameters). How does this change the expressible power of Excel spreadsheets?
- 10) Describe in 80 to 100 words which languages have the strangest features.

2. Question (20P)

Java 10 added the „var“ keyword. What (if any) effect does it have on the Java typing system? Make an educated guess why features like this were recently added to the Java language and not earlier.

3. Question (20P)

FORTH explicitly defines interpretation and compilation semantics for every word. E.g. the word „+“ adds two numbers from the stack when interpreted, however it adds machine code for that adds two numbers from the stack when compiled. Prefixing an operator by POSTPONE compiles the compilation semantics into a word being defined, i.e. when the word is executed, the operator is treated as if it was compiled. On the other hand the word immediate makes the compilation semantics of the last defined word equal to its interpretation semantics. Using this, explain why

: ENDIF POSTPONE THEN ; IMMEDIATE

makes ENDIF a synonym for THEN.

4. Question (20P)

To make a grammar LL-parsable, two techniques were presented in the lectures:

- Elimination of left recursion by replacing productions like $L \rightarrow L R_1 | L R_2 | L R_3 | R_4 | R_5 | R_6$ by $L \rightarrow R_4 L' | R_5 L' | R_6 L'$, $L' \rightarrow R_1 L' | R_2 L' | R_3 L' | \epsilon$
- Elimination of common starts (e.g. if / then vs. if / then / else) by $L \rightarrow R_1 R_2 | R_1 R_3$ by $L \rightarrow R_1 L', L' \rightarrow R_1 | R_2$

Discuss why these two techniques make the grammar LL-parsable without changing the set of created words.

5. Question (15P) *E_{extra}*

Explain the differences between C++ and Java references in 80 to 100 words

6. Question (15) *E_{extra}*

C++ uses pass-by-reference, but Ada, which is predominantly used by security sensitive programs uses pass-by-value-result. What are the similarities, what are the differences?