

# CPSC 449: Principles of Programming Languages

## Questions to prepare yourself for the Midterm

- Describe in not more than 4 sentences what steps are involved in a human being understanding a written word in a language
- Name 4 “items” humans want to communicate to each other using a language
- Give 3 examples why language is more than words.
- How does a language get born?
- Name 3 general problems with human languages.
- Name 3 different reasons why we have the Babel syndrome with regard to programming languages.
- Describe in not more than 4 sentences what steps are involved in a computer understanding a program
  - a) assuming that a compiler is used
  - b) assuming that an interpreter is used
- What do we need to define a programming language?
- Name the 3 general things that define (resp. influence) the semantics of a programming language.
- Name the two general problems that we have with programming languages.
- What is the basic idea of imperative programming languages?
- Give another name for the imperative programming paradigm.
- Name the 4 basic ideas of object-oriented programming languages
- Name 5 imperative programming languages
- Name 2 real object-oriented programming languages
- What was the first widely used programming paradigm?
- What was the second widely used programming paradigm?
- What are, in general, data types, resp. data structures?
- Name the build-in data types of Pascal
- Name the build-in (primitive) data types of Java
- Name 3 different primitive data types that occur in most imperative and object-oriented languages
- Name 4 structured data types in Pascal.
- Given a program fragment in Pascal (sequence of statements), provide the data type declarations needed to make this fragment work.
- What is the difference between arrays and files in Pascal?
- Given a particular set definition in Pascal, provide all possible values of the set data type.
- Given a particular pointer structure and a Pascal code fragment, draw a picture of the pointer structure after the fragment is executed.
- What is the general structure of a class definition in Java?
- What other way is there in Java to generate more complex data structures in addition to classes (and interfaces)?
- What is a Java interface?
- Given a program fragment in Java (from a method), provide the necessary class definitions to

make this fragment work.

- Name the three different kinds of control constructs.
- Given a case statement in Pascal (or a switch statement in Java), provide a Pascal (resp. Java) program fragment doing the same without using case (resp. switch).
- How is a block defined in Pascal?
- Name the different types of loop-constructs in Pascal.
- Why can't we use a for-loop to simulate every possible while-loop in Pascal?
- Given a particular program fragment with a loop, write a program fragment using a different loop to achieve the same effect
- Show that in Java every while loop can be simulated using a for-loop.
- How can you break out of a loop in Pascal?
- Name the ways how you can break out of a loop in Java.
- What conceptual differences are there between imperative programming and object-oriented programming with regard to control constructs?
- What is the difference between a variable in Pascal and a variable (for an object in a class) in Java?
- What is a buffer variable in Pascal?
- Discuss the similarities between a Pascal record type and a Java class with respect to accessing and manipulating data.
- What is the difference between a function and a procedure in Pascal?
- Provide some example Pascal functions/procedures, calls of these functions/procedures and results of these calls that show the difference between by-value parameters and by-reference parameters.
- What is the result of the new operator in Java?
- What is the difference between Java interfaces and abstract classes from the point of view of defining methods?
- Given some fragmental class/interface definitions in Java and a variable for an object of one of the classes, determine the methods (and their parameter types and result types) available for this object.
- What is overloading of a method or function name?
- What is overriding of a method?
- What is polymorphism?
- What is the general structure of a Pascal program?
- Name the declarations that occur in the declaration part of a Pascal program.
- Given a Pascal program fragment with indicated positions in it, explain to what parts of the fragment a variable name at the indicated positions refers to.
- What method in a collection of Java-files that constitute a program is called first?
- What are packages good for in Java?
- Is the import statement necessary to allow access to classes and methods in other packages?
- What is a shadowed variable in Java?
- Are there shadowed methods in Java?
- What is stored in an entry on a run-time stack?
- What has the Pascal run-time system to do when it removes the top element of the run-time stack?
- How does a run-time stack deal with recursive procedure calls?
- Are Java programs compiled or interpreted?

- What is a class template?
- Are there run-time stacks in the Java run-time system and if yes, what are they used for?
- What is the difference between a “normal” variable in Java and a class variable?
- Does the class variable concept fit into the object-oriented programming paradigm? Justify your answer!
- What is the task of a garbage collector?
- How does a garbage collector fulfill its task?
- What is the difference (from the point of view of a programmer) between an error and an exception in Java?
- What is the effect of the finally statement with regard to exception handling in Java?
- What is an event?
- What is an event listener in Java?
- Discuss the statement: “Exception handling is a special ability of object-oriented programming”!
- What data structure is used in Pascal for handling in- and output?
- Are Java streams similar to files, broader or more restricted? Justify your answer!
- What is a thread?
- What is the effect of the yield method?
- What method can be used by a thread to wait until another thread is finished?
- Discuss the differences of imperative and object-oriented programming with regard to
  - control flow
  - memory management
  - re-use potential
- What is the basic idea of functional programming languages?
- What is usually seen as the most primitive functional programming language?
- What are the two cornerstones of the  $\lambda$ -calculus?
- How can functions with several arguments be expressed by functions with only one argument?
- Define the set of lambda-expressions  $\text{Lambd}$ .
- Name the two rules in the  $\lambda$ -calculus that are used to evaluate lambda-expressions.
- Given two lambda-expressions, show that they are within the  $\equiv$ -relation
- Given a natural number, give the Church-integer presentation of it.
- What is the definition of the Y combinator in  $\lambda$ -calculus?
- What is a fixed point of a function?
- Given a set of clauses, use the Davis-Putnam method (as described for the assignment) to show that the set is unsatisfiable or to produce a model for it.
- Given a tree produced by the Davis-Putnam method, provide a set of clauses that allows to produce exactly this tree.