Concepts in Programming Languages Past Paper

- Adapted from Revision Guide (revised 2017)

I. Introduction and motivation

- Programming-language concepts, design, methods, paradigms, influences; Application domains; Execution models; Foundations; Standardisation.
- 2006 Paper 6 Question 7 (a)
- 2012 Paper 3 Question 6 (a)
- 2015 Paper 3 Question 5 (a, b)

II. FORTRAN: A simple procedural language

- FORTRAN 77; Execution model; Compilation; Data types; Control structures; Syntax; Types; Storage; Aliasing; Parameters.
- 2006 Paper 6 Question 7 (b)
- 2007 Paper 5 Question 7 (a)
- 2007 Paper 6 Question 7 (c)
- 2009 Paper 3 Question 2 (a)
- 2010 Paper 3 Question 5 (a)

III. LISP: Functions, recursion, and lists

- LISP; Programming-Language phrases; S-expressions; quote; Static and Dynamic scope; Abstract machine; Recursion; Garbage collection; Programs as data; Reflection; Parameter passing.
- 2006 Paper 6 Question 7 (c)
- 2007 Paper 6 Question 7 (a)
- 2009 Paper 3 Question 2 (a)
- 2014 Paper 3 Question 6 (a)
- 2007 Paper 5 Question 7 (b)
- 2008 Paper 6 Question 7 (a)
- 2011 Paper 3 Question 6 (a (ii))

IV. Block-structured procedural languages – Algol and Pascal

- Parameters; Parameter-passing; Block structure; Algol 60; Recursion; Stack; Type system; Algol 68; BNF syntax; Heap; Garbage collection; Pascal; Quasi-strong typing; Variant records.
- 2006 Paper 6 Question 7 (b)
- 2007 Paper 6 Question 7 (b)
- 2008 Paper 5 Question 7 (a)
- 2010 Paper 3 Question 5 (a)
- 2012 Paper 3 Question 6 (c)
- 2013 Paper 3 Question 6 (b)
- 2007 Paper 5 Question 7 (c)

- 2008 Paper 6 Question 7 (b)
- 2009 Paper 3 Question 2 (c)
- 2011 Paper 3 Question 6 (a (i))
- 2013 Paper 3 Question 6 (a (i))
- 2015 Paper 3 Question 5 (c)

V. Object-oriented languages – SIMULA and Smalltalk

- Objects in SML; Dynamic lookup; Abstraction; Subtyping; Inheritance; Subtyping vs. inheritance; SIMULA; Classes, objects and activation records; Subclasses and inheritance; Type checking and subtyping; Smalltalk; Dynabook; Syntax; Abstraction; Messages; Methods; Instance variables; Interfaces as types; Subtyping.
- 2006 Paper 6 Question 7 (d)
- 2008 Paper 5 Question 7 (c)
- 2011 Paper 3 Question 6 (a)
- 2013 Paper 3 Question 6 (a (ii))
- 2007 Paper 6 Question 7 (d)
- 2010 Paper 3 Question 5 (b)
- 2012 Paper 3 Question 6 (f)

VI. Types in programming languages

- Types; Type systems; Type safety; Type checking; Static vs. dynamic type checking; Type checking in SML; Type equality; Type declarations; Type inference; Type inference in SML; Polymorphism; let-polymorphism; Polymorphic exceptions.
- 2008 Paper 5 Question 7 (b)
- 2010 Paper 3 Question 5 (c)
- 2012 Paper 3 Question 6 (b)
- 2013 Paper 3 Question 6 (c)
- 2015 Paper 3 Question 5 (d, e, f)
- 2009 Paper 3 Question 2 (b)
- 2011 Paper 3 Question 6 (b)
- 2012 Paper 3 Question 6 (e)
- 2014 Paper 3 Question 6 (c)
- 2016 Paper 3 Question 5 (d, e)

VII. Scripting Languages – JavaScript

- Scripting vs. dynamic typing; JavaScript; Prototypal inheritance; Browser integration. Students might consider the following questions:
- “Scripting languages and dynamically typed languages are identical; discuss”
- “Discuss the notion of ‘class’ in relation to JavaScript”

VIII. Data abstraction and modularity – SML Modules

- Modules language; Signatures; Structures; Concrete and opaque signatures; Signature inclusion; Signature matching; Subtyping; Information hiding; Functors.

- 2007 Paper 5 Question 7 (d)
- 2010 Paper 3 Question 5 (d)
- 2013 Paper 3 Question 6 (d)
- 2009 Paper 3 Question 2 (d)
- 2011 Paper 3 Question 6 (c)
- 2014 Paper 3 Question 6 (d)

IX. Languages for concurrency and parallelism.

- Theoretical models; Threads, shared memory, message passing; Distributed memory, multi-core, cloud computing; Programming-language support for parallelism and distribution. Internal and external iteration.
- 2014 Paper 3 Question 6 (b)

X. Functional-style programming meets object-orientation.

- Scala and Java 8; Procedural programming; Declarative programming; Mutable state; Blocks; Functions; Parameter passing; Classes and objects; abstract classes; traits; case classes; Pattern matching; Generic types and methods; Variance annotations; Functions as objects;
    - [No longer explicitly lectured:] Type parameter bounds; View bounds; Implicit parameters; Implicit conversions; Mixin-class composition.
- 2008 Paper 6 Question 7 (c)
- 2010 Paper 3 Question 5 (e)
- 2012 Paper 3 Question 6 (d)
- 2009 Paper 3 Question 2 (e)
- 2011 Paper 3 Question 6 (d)
- 2013 Paper 3 Question 6 (e)

XI. Miscellaneous concepts Keywords:

- Monads, GADTs, Reified continuations, Dependent typing.
- 2016 Paper 3 Question 5 (a, b, c)