University Computer Science

Syllabus Checklist ·

Tick one box per statement: ☐ not yet ☐ nearly ☐ confident. / /

1. Programming Fundamentals

1.1 Values, types and input/output 、 /

☐☐☐ Values, types and input/output

1.2 Selection and conditionals

☐☐☐ Selection and conditionals

1.3 Iteration

☐☐☐ Iteration

1.4 Functions

☐☐☐ Functions

1.5 Collections

☐☐☐ Collections

1.6 Files and errors

☐☐☐ Files and errors

1.7 Decomposition, testing and debugging 、

☐☐☐ Decomposition, testing and debugging

2. Discrete Mathematics

2.1 Logic and proofs

☐☐☐ Logic and proofs

2.2 Sets, functions and relations 、

☐☐☐ Sets, functions and relations

2.3 Induction and recursion

☐☐☐ Induction and recursion

2.4 Counting and combinatorics

☐☐☐ Counting and combinatorics

2.5 Graphs and trees

☐☐☐ Graphs and trees

2.6 Discrete probability

☐☐☐ Discrete probability

3. Computer Systems Fundamentals

3.1 Binary and data encoding

☐☐☐ Binary and data encoding

3.2 Logic gates and digital building blocks

☐☐☐ Logic gates and digital building blocks

3.3 Processor and memory overview

☐☐☐ Processor and memory overview

3.4 Machine instructions

☐☐☐ Machine instructions

3.5 The operating-system interface

☐☐☐ The operating-system interface

3.6 Compilation and execution path

☐☐☐ Compilation and execution path

4. Object-Oriented Programming

4.1 Objects and classes

☐☐☐ Objects and classes

4.2 Encapsulation

☐☐☐ Encapsulation

4.3 Composition and inheritance

☐☐☐ Composition and inheritance

4.4 Interfaces and polymorphism

☐☐☐ Interfaces and polymorphism

4.5 Exceptions and error handling

☐☐☐ Exceptions and error handling

4.6 Testing and design quality

☐☐☐ Testing and design quality

5. Data Structures and Algorithms

5.1 Asymptotic analysis

☐☐☐ Asymptotic analysis

5.2 Lists, stacks and queues 、

☐☐☐ Lists, stacks and queues

5.3 Hashing

☐☐☐ Hashing

5.4 Trees and heaps

☐☐☐ Trees and heaps

5.5 Graphs and traversal

☐☐☐ Graphs and traversal

5.6 Sorting

☐☐☐ Sorting

5.7 Searching

☐☐☐ Searching

6. Computer Architecture

6.1 Instruction set architecture

☐☐☐ Instruction set architecture

6.2 Datapath and control

☐☐☐ Datapath and control

6.3 Pipelining

☐☐☐ Pipelining

6.4 Memory hierarchy

☐☐☐ Memory hierarchy

6.5 Input/output /

☐☐☐ Input/output

6.6 Parallelism and performance

☐☐☐ Parallelism and performance

7. Operating Systems

7.1 Processes and threads

☐☐☐ Processes and threads

7.2 Scheduling

☐☐☐ Scheduling

7.3 Synchronisation

☐☐☐ Synchronisation

7.4 Virtual memory

☐☐☐ Virtual memory

7.5 File systems

☐☐☐ File systems

7.6 Protection and security

☐☐☐ Protection and security

8. Databases

8.1 Relational modelling

☐☐☐ Relational modelling

8.2 Relational algebra and SQL SQL

☐☐☐ Relational algebra and SQL

8.3 Normalisation

☐☐☐ Normalisation

8.4 Indexes and query plans

☐☐☐ Indexes and query plans

8.5 Transactions and concurrency

☐☐☐ Transactions and concurrency

8.6 Security and data governance

☐☐☐ Security and data governance

9. Computer Networks

9.1 Layering and architecture

□□□ Layering and architecture

9.2 Link layer and IP IP

□□□ Link layer and IP

9.3 Routing

□□□ Routing

9.4 Transport (TCP/UDP) (TCP/UDP)

□□□ Transport (TCP/UDP)

9.5 Application protocols

□□□ Application protocols

9.6 Network security

□□□ Network security

9.7 Distributed systems fundamentals

□□□ Distributed systems fundamentals

10. Algorithm Design and Analysis

10.1 Recurrences and analysis tools ⟦number⟧

□□□ Recurrences and analysis tools

10.2 Divide and conquer

□□□ Divide and conquer

10.3 Greedy algorithms

□□□ Greedy algorithms

10.4 Dynamic programming

□□□ Dynamic programming

10.5 Graph optimisation

□□□ Graph optimisation

10.6 Reductions and NP-completeness NP

□□□ Reductions and NP-completeness

11. Theory of Computation and Compilers ⟦number⟧

11.1 Automata and regular languages

□□□ Automata and regular languages

11.2 Context-free languages and parsing theory

□□□ Context-free languages and parsing theory

11.3 Computability

□□□ Computability

11.4 Lexical analysis

□□□ Lexical analysis

11.5 Parsing ⟦number⟧

□□□ Parsing

11.6 Semantic analysis and code generation

□□□ Semantic analysis and code generation

12. Software Engineering

12.1 Requirements

□□□ Requirements

12.2 Architecture and design

□□□ Architecture and design

12.3 HCI and accessibility

□□□ HCI and accessibility

12.4 Version control and teamwork

□□□ Version control and teamwork

12.5 Testing and continuous integration

□□□ Testing and continuous integration

12.6 Security and professional ethics

□□□ Security and professional ethics

12.7 Team project

□□□ Team project

13. Artificial Intelligence and Machine Learning

13.1 Search

☐☐☐ Search

13.2 Knowledge and reasoning

☐☐☐ Knowledge and reasoning

13.3 Probability models

☐☐☐ Probability models

13.4 Regression and classification

☐☐☐ Regression and classification

13.5 Optimisation and learning

☐☐☐ Optimisation and learning

13.6 Evaluation and leakage

☐☐☐ Evaluation and leakage

13.7 Fairness and limitations

☐☐☐ Fairness and limitations