



Habib University
shaping futures

Creative | Collaborative | Cutting-edge

BS

**COMPUTER
SCIENCE**

BS IN

COMPUTER SCIENCE

FACULTY MEMBERS

Alam, Shah Jamal, Assistant Professor, Khidmat Adviser, Registrar

Hussain, Shahid, Assistant Professor

Khan, Umair Azfar, Assistant Professor, Entrepreneurship Adviser

Rashid, Jibran, Assistant Professor, Higher Education Adviser

Raza, Syeda Saleha, Assistant Professor, Industry Adviser

Saif Ur Rahman, Syed, Assistant Professor

Saleem, Waqar, Assistant Professor and Program Director

Computer Science is the study of computation - what can and cannot be computed, how can computation be made more efficient, how to build machines that can compute, and which spheres of human activity can benefit from computational approaches. It is deeply rooted in logic and mathematics. Theoretical Computer Scientists are constantly pushing the frontiers of computation by inventing new computational approaches. Practical Computer Scientists apply the theory of Computer Science to different application areas like science, finance, medicine, entertainment, education, communication, engineering, art, and the humanities.

PROGRAM VISION

The Computer Science program educates students in the theory, systems, and applications of Computer Science in order to enable them to make impactful contributions to society and prepares them for success in entrepreneurship, higher education, and industry.

DEGREE REQUIREMENTS(MAJOR)

In addition to the 11 units required by the University for every student, Computer Science majors complete 35.5 units in the following categories:

- Foundation, 4 units
- Kernel, 4 units
- Electives, 6 units
- Career Development, 2 units
- Support, 10.5 units
- University Electives, 6 units
- Capstone Project, 3 units

Habib Liberal Core (described on Page 33) contributes 11 units as follows. Each course from Habib Liberal Core including the local language requirement contributes 1 unit. The Creative Practice requirement of Habib Liberal Core is fulfilled by

- CS290 Khidmat, 1 unit

which is to be completed in consultation with the Khidmat adviser.

Computer Science Foundation comprises courses that provide incoming students the necessary skills in order to proceed in the Computer Science major. The courses are:

- CS110 Computational Thinking I, 1 unit
- CS111 Computational Thinking II, 1 unit
- CS130/EE171 Digital Logic and Design, 1 unit
- CS132 Elements of Computing Systems, 1 unit

Computer Science Kernel comprises essential courses on the theory and implementation of Computer Science. The courses are:

- CS200 Functional Data Structures, 1 unit
- CS212 Nature of Computation, 1 unit
- CS200 Object Oriented Programming and Design Methodologies, 1 unit
- CS310 Design and Analysis of Algorithms, 1 unit

Computer Science Electives provide students the choice to explore Computer Science in its breadth and depth. The program offers courses fulfilling each of 3 categories: theory, systems, and application. Students must complete 6 units with at least 1 unit in each category.

Computer Science Support develops the supporting approaches and skills necessary for a Computer Science graduate. The courses are:

- MATH101 Calculus I, 1 unit
- MATH102 Calculus II, 1 unit
- MATH201 Differential Equations, 1 unit
- MATH210 Probability and Statistics, 1 unit
(may be substituted by EE352 Probability and Stochastic Processes)
- MATH205 Linear Algebra, 1 unit
- EE111 Electric Circuits, 1 unit
- ENGR291 Engineering Workshop, 0.5 units

In addition, students must complete 4 units as follows

- 2 units of Natural Science courses from which at least one course includes a lab component.
- 2 units of courses fulfilling the Technical Skills category.

University Electives allow Computer Science majors to develop breadth outside their major. Students must complete 6 units as follows.

- 1 unit in Electrical Engineering
- 1 unit in Arts, Humanities, and Social Sciences
- 2 units in Digital Humanities
- 2 units from any courses on offer at the University.

Career Development courses prepare students for success in life beyond University by providing essential skills. Students must complete 2 units including

- CS290 Technical Communication

The remaining 1 unit is to be completed in consultation with either of the 3 track advisers: Entrepreneurship, Research, and Industry.

Capstone Project is a year long project attempted in the final year of studies at the University. It comprises the courses

- CS490 Kaavish I, 1.5 Units
- CS491 Kaavish II, 1.5 Units

The project must be carried out in consultation with either of the 3 track advisers: Entrepreneurship, Research, and Industry.

4 MINOR REQUIREMENTS

The Computer Science program offers a minor to students from other programs who wish to develop a significant grounding in Computer Science. The minor comprises 8 units defined as follows:

- 3 units from Computer Science Foundation that must include CS110 and CS111.
- Any 3 units from Computer Science Kernel.
- Any 2 units from Computer Science Electives.

5 GRADUATION GPA REQUIREMENT (MIN)

In order to qualify for a CS major or minor, a student must meet the University's minimum GPA requirement. In addition, he/she must have at least a C+ grade in each Computer Science Foundation and Computer Science Kernel course attempted and at least a C- in every other Computer Science course attempted.

6 COURSE DESCRIPTIONS FOR FALL 2016 AND SPRING 2017

CS351 Artificial Intelligence

Prerequisite: CS200 Functional Data Structures

Note:

- Fulfils CS Applications Elective.

Study of the techniques used in the science and engineering of making machines intelligent; techniques include: problem solving, decision making, learning, planning, and reasoning; application areas explored are computer vision, robotics, and deep learning; specific topics include: intelligent search techniques, games, and adversarial search using minimax and alpha-beta pruning, supervised learning via decision trees, naïve Bayes, and artificial neural networks, K-means clustering, reasoning via first-order logic, Bayesian networks, and evolutionary algorithms.

CS412 Algorithms: Design and Analysis

Prerequisite:

- CS200 Functional Data Structures
- CS212 Nature of Computation

Note:

- Part of CS Kernel.
- Listed earlier as CS310 Design and Analysis of Algorithms.

Develops tools and techniques that aid in designing correct, efficient algorithms for computational problems and analyzing their correctness and running time; some of the discussed techniques are: greedy method, divide-and-conquer, dynamic programming, hashing, randomization, network flows, linear programming, fast Fourier transform, and techniques for thinking about solving problems in parallel; analysis tools include: recurrences, probabilistic analysis, amortized analysis, and potential functions.

CS317 Combinatorial Machine Learning

Co-requisite/Prerequisite:

- CS412 Algorithms: Design and Analysis

Note:

- Fulfils CS Theory Elective.

Examines decision trees and rules widely used in supervised machine learning as combinatorial objects; investigates algorithms to construct and optimize these objects using both dynamic programming and greedy heuristics; topics include: decision tables, decision trees, decision rules, tests (reducts), supervised learning, and algorithms for the construction of trees, rules, and tests.

CS110 Computational Thinking I

Prerequisite: None

Note:

- First in a 2-course sequence: CS110 and CS111
- Part of CS Kernel.

Explores the historical context of Computer Science; highlights the application of computing in solving everyday problems; develops familiarity with algorithmic thinking and applications; introduces basic data structures; cultivates problem solving and debugging skills; focuses on the use of related formalism and mathematics; develops expertise in programming in a high level language; specific topics include: data structures (LIFO/FIFO, trees, graphs) and related algorithms, algorithm analysis, asymptotic notation, divide and conquer algorithms, Master theorem, sorting, functions, recursion, pointers, structs, and classes.

CS111 Computational Thinking II

Prerequisite: CS110 Computational Thinking I

Note:

- Second in a 2-course sequence: CS110 and CS111
- Part of CS Foundation.

Explores the historical context of Computer Science; highlights the application of computing in solving everyday problems; develops familiarity with algorithmic thinking and applications; introduces basic data structures; cultivates problem solving and debugging skills; focuses on the use of related formalism and mathematics; develops expertise in programming in a high level language; specific topics include: data structures (LIFO/FIFO, trees, graphs) and related algorithms, algorithm analysis, asymptotic notation, divide and conquer algorithms, Master theorem, sorting, functions, recursion, pointers, structs, and classes.

CS224 Object Oriented Programming and Design Methodologies

Prerequisite: CS110 Computational Thinking I

Note:

- Part of CS Kernel (as of 2016).
- Fulfills CS Logic, Languages, and Programming Elective (prior to 2016).

Introduces object oriented and related memory concepts; motivates C++ as the language of choice; topics include: pointers and structs, objects, heap allocation, data encapsulation, classes, namespaces, constructors and destructors, virtual functions and destructors, operator overloading and standard input/output, inheritance and polymorphism, templates, standard library containers, and software design using UML 2.0.

CS261 Understanding Social Networks

Prerequisite: Sophomore level

Note:

- Fulfills CS Applications Elective.

Introduces the theory and methods for social network analysis coming from sociology, communications studies, and mathematics and computer science; explores real-world networks from a variety of domains such as online social networks (e.g., Facebook and Twitter), political networks, informal money transfer, kinship and sexual networks; introduces methods and computational tools to analyze such social networks using modern software.

CS200 Functional Data Structures

Prerequisite: CS111 Computational Thinking II

Note:

- Part of CS Kernel.

A hands-on exploration of common data structures using functional programming; explores the use of data structures in popular software; introduces functional programming which is used to implement the covered data structures; these include: binary search tree (BST), randomized binary tree, treap, self-balancing trees (AVL, B-tree, red-black), heap, linked lists (single, double, curricular), skip list, hash table, and graph.

CS130 Digital Logic and Design

Prerequisite: None.

Note:

- Satisfies the Systems requirement of CS Foundation.
- Cross-listed with EE171.

Introduction to the design of digital hardware, realization of computation with logic gates; Boolean algebra, design of combinational logic circuits and analysis and design of clocked sequential logic circuits, circuits for arithmetic operations; introduction to hardware description language and its application to logic design.

CS132 Elements of Computing Systems

Prerequisite: CS111 Computational Thinking II

Note:

- Satisfies the Systems requirement of CS Foundation.

A hands-on exploration of the construction of a modern, full-scale computer system from the ground up; exposure to major Computer Science abstractions studied in detail in later courses; 12 implementation projects focusing on building the hardware platform and software hierarchy of a computer system.

CS212 Nature of Computation

Prerequisite: CS111 Computational Thinking II

Note:

- Satisfies the Theory requirement of CS Kernel.

Investigation of fundamental challenges at the frontiers of theoretical Computer Science; establishes basic mathematical preliminaries and explores historical breakthrough results in Computer Science; provides practice in developing a rigorous mathematical argument and a clear intuitive explanation of a complex argument; specific topics include: proofs, automata and grammars, Turing machines and the halting problem, Godel's completeness and incompleteness theorems, computational complexity, interactive and zero-knowledge proofs, and quantum computing.

CS355 Databases

Prerequisite: CS111 Computational Thinking II

Note:

- Fulfills CS Applications Elective (as of 2016).
- Fulfills CS Software Systems Elective (prior to 2016).

A detailed exploration of the theoretical and practical aspects of Relational Database management Systems (RDBMS); develops an understanding of database modeling,

relational algebra, structured query language (SQL), components of Database Management System (DBMS), transaction management and concurrency control, database fine-tuning via indexing and partitioning, and database connectivity with front-end applications; discusses administrative aspects of database systems including database security, database management vs data warehousing vs data mining, and big data and its challenges.

Course Category	Number of Courses to complete		
	Total	Compulsory	Choice
University Requirements			
Habib Liberal Core (including Creative Practice)	11	9	2
Computer Science			
Foundation	4	4	
Core/Kernel	4	4	
Theory Elective	1		1
Systems Elective	1		1
Applications Elective	1		1
CS Free Electives	3	1	2
Capstone Project	3	3	
Khidmat	1	1	
Support			
Mathematics	5	5	
Career Development**	2		2
Science & Engineering	3.5	3.5	
Electives			
Arts, Humanities and Social Sciences	1		1
School of Science & Engineering	1		1
Digital Humanities	2		2
Free Electives	2		2
Overall	43.5	30.5	13

CS Khidmat and one CS elective cover two area of requirement and is not calculated twice in overall total units:

**One CS free elective counts toward fulfillment of one Career Development course



HABIB UNIVERSITY:
UNIVERSITY AVENUE, OFF SHAHRAH-E-FAISAL,
GULISTAN-E-JAUHAR, KARACHI

For complete course description and department requirements,
see Habib's online course catalog at www.habib.edu.pk

<https://habib.edu.pk/academics/sse/computer-science/>

 www.habib.edu.pk

 admissions@habib.edu.pk

 +92 21 11 10 HABIB (42242)

 [HabibUniversity](https://www.facebook.com/HabibUniversity)