

Ansøgningsskabelon

Oplysninger fra CVR

CVR-nummer

29057753

Virksomhedens navn

IT-Universitetet i København

Virksomhedens adresse

Rued Langgaards Vej 7, 2300 København S

0. Ansøgningens type

Der ansøges om

- ☐ Ansøgningsskema 1 bacheloruddannelse 180 ECTS-point
- ☐ Ansøgningsskema 2 kandidatuddannelse 120 ECTS-point
- ☐ Ansøgningsskema 3 kandidatuddannelse 75 ECTS-point
- ☐ Ansøgningsskema 4 erhvervskandidatuddannelse 75-120 ECTS-point
- ☒ Ansøgningsskema 5 erhvervskandidatuddannelse som kombinationsuddannelse (1+2) 120 ECTS-point
- ☐ Ansøgningsskema 6 erhvervskandidatuddannelse som kombinationsuddannelse 75-120 ECTS-point

1. Grundoplysninger for uddannelsen

Navn på institution

IT-Universitetet i København

Udbudsted

2300 København S

Angiv evt. supplerende bynavn

Uddannelsstype

Erhvervskandidat som kombinationsuddannelse

Fagbetegnelse dansk

Avanceret software engineering

Fagbetegnelse engelsk

Advanced Software Engineering

Betegnelse dansk (titel)

candidatus/candidata scientiarum (cand. scient.) i avanceret software engineering

Betegnelse engelsk (titel)

Master of Science (MSc) in Advanced Software Engineering

Hvilket hovedområde hører uddannelsen under?

Naturvidenskab

Vælg undervisningssprog

- ☐ Dansk
- ☒ Engelsk

Forslag til censorkorps. Angiv hvilket censorkorps, som institutionen ønsker at knytte til uddannelsen

Censorkorps for datalogi

Hvilke adgangskrav gælder til uddannelsen?

Adgangskrav

- En universitetsbacheloruddannelse eller en professionsbacheloruddannelse, der relaterer sig til datalogi, softwareudvikling, computer engineering, robotteknologi eller tilsvarende.
- Ved anden uddannelse (f.eks. en international uddannelse), der kan sidestilles med en universitetsbacheloruddannelse eller en professionsbacheloruddannelse, vil IT-Universitetet foretage en individuel vurdering af den pågældende uddannelse.
- Engelsk på B-niveau med minimum 3 i snit.

Specifikke adgangskrav:

1. Grundig forståelse og beherskelse af programmering, herunder imperative og objektorienterede programmeringsparadigmer, og erfaring med et programmeringssprog i disse kategorier (30 ECTS)
2. Grundlæggende principper i software engineering (5 ECTS)
3. Grundlæggende matematik, herunder diskret matematik og statistik (5 ECTS)
4. Principper for databasesystemer (5 ECTS)
5. Programmeringssprog og deres semantik samt grundlæggende syntaksanalyse (5 ECTS)
6. Kendskab til operativsystembegreber, herunder ressourceadministration, parallelprogrammering, og distribuerede systemer (5 ECTS)
7. Grundlæggende algoritmik (5 ECTS)

Ansøgere skal opfylde alle adgangskrav (1-4) og mindst to af adgangskravene (5-7)

Bacheloruddannelser der giver direkte adgang:

- Softwareudvikling (ITU)
- Datalogi (AAU)
- Software (AAU)
- Datalogi (AU)
- Datalogi (KU)
- Softwareteknologi (DTU) - Bachelor of Science in Engineering (Software Technology)
- Datalogi (SDU)
- Software Engineering (SDU-Odense)
- Software Engineering (SDU-Sønderborg)

Beskriv uddannelsens formål og erhvervssigte

Dimittender fra kandidatuddannelsen i Advanced Software Engineering vil erhverve sig praktiske kompetencer sammenlignelige med, eller stærkere end dem, som datalogikandidater fra danske og udenlandske universiteter besidder. Uddannelsen giver de studerende mulighed for at skræddersy deres tekniske fokus samtidig med, at de erhverver vigtige ledelseskompetencer. Dette forbereder dimittenderne til arbejdsroller, der både kræver dybe tekniske kompetencer og ledelseskompetencer, hvilket gør det muligt for dem at bidrage i en bred vifte af professionelle miljøer.

Kandidatuddannelsen i Advanced Software Engineering sikrer at alle dimittender:

- Har en grundig forståelse af programmeringsparadigmer og hurtigt kan beherske nye programmeringssprog og udviklingsplatforme.
- Kan anvende principper fra software engineering til at designe, planlægge og lede softwareudviklingsprojekter.
- Kender og kan anvende moderne teknikker til at vedligeholde, videreudvikle og vurdere kvaliteten af softwaresystemer.
- Kan idriftsætte softwareløsninger.

Givet viften af valgfrie kurser kan dimittenderne endvidere erhverve viden om it-sikkerhed og principper for forretnings- og teknologiledelse.

Angiv hvilken struktur og konstituerende faglige elementer uddannelsen består af.

Se uploaded bilag "Uddannelsens struktur og konstituerende faglige elementer"

Angiv antal ECTS-point for hele uddannelsen

120

Angiv normeret studietid i mdr.

34

Her angives beskæftigelseskravet i timer pr. uge (årsnorm)

25

Angiv antal ECTS-point på heltidsforløb (med SU)

60

Heltidskandidatuddannelse som erhvervskandidatuddannelsen giver ret til indskrivning på

Master of Science (MSc) in Advanced Software Engineering (se øvrige kommentarer)

Angiv den firecifrede UDD-kode fra Danmarks Statistik, som er tilknyttet den heltidskandidatuddannelse som erhvervskandidatuddannelsen giver ret til indskrivning på

0000

2. Øvrige krav til ansøgningen

Kort beskrivelse af hvordan aftagere og øvrige relevante interessenter er inddraget i processen omkring udvikling af uddannelsen.

Aftagerinddragelse

I forbindelse med udviklingen af erhvervskandidatuddannelsen i Advanced Software Engineering har både IT-Universitetets egne aftagerpaneler samt et panel bestående af udvalgte og relevante aftagere fra såvel private som offentlige virksomheder og organisationer været med i udviklingsprocessen og givet input og feedback til uddannelsen. Desuden har panelerne forholdt sig til, om de ville være interesserede i at ansætte studerende og kandidater fra uddannelsen.

Proces for inddragelse

30.09.2024: Aftagerpanel for de eksisterende uddannelser BSc i Data Science, BSc i Software Development og MSc i Data Science, MSc i Computer Science, MSc i Software Design gav på deres halvårige møde input og feedback til forslag til uddannelsens indhold, struktur og relevans.

25.11.2024: Fysisk møde på ITU med en udvalgt gruppe af aftagere, der gav feedback og forslag til første udkast til uddannelsens programme design.

Aftagerne var repræsentanter fra følgende organisationer og virksomheder: Lego, SimCorp, Saxo Bank, CGI Danmark, Salesforce, Novo Nordisk, Netcompany, Nordea, Erhvervsstyrelsen.

Panelet gav både input til indhold og struktur i forhold til, at de studerende skal være ansat sideløbende med at de tager uddannelsen. De forholdt sig også konkret til om deres virksomhed eller organisation ville ansætte studerende fra uddannelsen.

Marts-august 2025: møder med enkelte større relevante aftagere om samarbejde og ansættelse af optagne studerende på uddannelsen.

Intern inddragelse

I perioden forår 2024 frem til januar 2025 er forslag til uddannelsen blevet udviklet med input, feedback og høring i alle dele af organisationen omfattende et større antal råd, nævn og grupperinger for uddannelse og forskning såsom studienævn, Subject Area Teams, uddannelsesledere, forskningsgrupper, institutter, direktion og bestyrelse.

3. Baggrundsuplysninger til brug for sagsbehandling

Henvisning til evt. tidligere godkendt heltidsuddannelse og dennes takstindplacering

Forslag til takstindplacering

Heltidstakst 3

Begrundelse for takstindplacering

Da uddannelsens kernefaglighed hører under det naturvidenskabelige område, foreslår IT-Universitetet, at uddannelsen indplaceres på takst 3 i lighed med øvrige datalogiske uddannelser.

Beskriv ligheder og forskelle til beslægtede uddannelser, herunder beskæftigelse og eventuel dimensionering

Til forskel fra ellers beslægtede uddannelser (se listen nedenfor), tilbyder kandidatuddannelsen i Advanced Software Engineering (ASE), via sin 1+2 opbygning, en enestående kombination af akademisk ekspertise i software engineering med praktisk industrierfaring. Praktisk industrierfaring er et integreret kerneelement i uddannelsen fra første år. Dette sikrer at de studerende systematisk erhverver erfaring med software engineering i en erhvervsmæssig kontekst, og dermed at dimittenderne er udrustet til umiddelbart og effektivt at bidrage i deres kommende jobfunktioner.

Helt fra første semester samarbejder de studerende med erhvervspartnerne i projektkurser, som giver tidligt indblik og engagement i realistiske problemstillinger og kontekster. Startende med uddannelsens andet år har de studerende lønnet arbejde i virksomheder, anvender deres evner og kompetencer i praksis, og har fleksible muligheder for at skræddersy deres specialisering til skiftende industribehov.

Kandidatuddannelsen i Advanced Software Engineering giver dimittenderne de kompetencer der er nødvendige for at designe, udvikle og lede komplekse softwaresystemer og –projekter i praksis, i stedet for udelukkende at fokusere på teori og forskning. Listen nedenfor viser en række andre danske kandidatuddannelser, hvis akademiske indhold og kompetencer overlapper med kandidatuddannelsen i Advanced Software Engineering, og afspejler arbejdsmarkedets stærke efterspørgsel efter ekspertise i disse områder. Kandidatuddannelsen sætter rammerne for at erhverve både de akademiske og praktiske kompetencer, der er nødvendige, for at imødekomme dette behov.

Vi har identificeret følgende uddannelser som teknisk-indholdsmæssigt er beslægtede med kandidatuddannelsen i Advanced Software Engineering, men adskiller sig fra

den ved fraværet af direkte og praktisk industrikontakt, som beskrevet ovenfor: MSc in Computer Science and Engineering (DTU), MSc in Computer Science (RUC), MSc in Software Engineering, MSc in Software Engineering (industrial master), Master of Science in Engineering (MSc) / Sønderborg (SDU), MSc in Engineering-Software Engineering specialization (AU), MSc in Software Engineering (AAU) and Master in Computer Science (ITU)

Vælg forventet opstartsår

2027

Angiv forventet optag på uddannelsen det første år

18

Angiv forventet optag på uddannelsen det andet år

42

Angiv det forventede optag på uddannelsen det tredje år

50

Øvrige bemærkninger til ansøgningen?

I forhold til tilbagefalds-fuldtidsuddannelse: Der henvises til ansøgning oktober 2025, om at udbyde denne 1+2 kombinationsuddannelse i Advanced Software Engineering som heltidsuddannelse. Heltidsuddannelsen vil kun have indskrivning/optag som tilbagefaldsmulighed.

☒ Det bekræftes hermed at ansøgningen er godkendt af rektor

Navn på kontaktperson

Mette Holm Smith

Kontaktpersons e-mail

meho@itu.dk

Kontaktpersonens telefonnummer

72185087

Vedhæft samlet PDF-fil

ASE 1+2 Uddannelsens struktur og konstituerende faglige elementer.pdf

269 KB

Uddannelsens struktur og konstituerende faglige elementer

The description below is written in English as the programme is an English taught programme and all course titles and content descriptions are “born” in English.

Program Structure

The study activities in the program and their organization over the three years are depicted in the following table and composed of 105 ECTS constituent elements: 45 ECTS mandatory courses, 30 ECTS constituent electives and 30 ECTS thesis. Furthermore 15 ECTS elective study activities.

Semester 1	Practical Software Analysis	Introduction to Machine Learning	Practical concurrent and parallel programming	Constituent Elective*
Semester 2	Modelling Systems and Languages	Software Architecture	DevOps, Software Evolution and Software Maintenance	Constituent Elective*
Semester 3	Constituent Elective*	Constituent Elective*	Company 50%	
Semester 4	Elective*	Elective*	Company 50%	
Semester 5	Thesis (30 ECTS)		Company 50%	
Semester 6			Company 50%	

* The electives can be used for writing a project under supervision for up to 30 ECTS. At least one of the electives in the first year should be a project course.

Constituent mandatory elements:

Every student must take in total 6 mandatory courses (45 ECTS). These courses cover topics identified as core knowledge areas essential for graduates of the MSc in Advanced Software Engineering. They focus on providing students with the technical competencies required to apply and develop software engineering techniques that meet the envisioned needs of the industry. The six mandatory courses in the program are:

- **Practical Software Analysis.** This course teaches skills required to develop techniques and tools for automatically analyzing software systems. Topics include software testing (e.g., regression testing), symbolic execution, and fuzzing. These methods are widely employed in modern software development to ensure software quality. The course emphasizes the application of these techniques to medium- to large-scale applications. To provide hands-on experience, students apply these techniques to realistic applications in a project as a part of course activities.

- **Modelling Systems and Languages.** This course focuses on developing skills for working with software in an abstract and semi-automated fashion, as in a production process (as opposed to a hand crafting process). The course introduces modeling languages and models, as first-class artifacts, that are designed, manipulated and translated to code in an automatic fashion. Students learn to design, evaluate, and implement new languages through hands-on experience, and use languages to describe systems concisely and with fewer errors.
- **Introduction to Machine Learning.** This course introduces Machine Learning (ML) techniques and relevant mathematical foundation. Machine learning proficiency is increasingly vital in software development processes, both in creating new ML methods and applying them contextually. The course covers the fundamentals of machine learning models, including their development, training, evaluation, and implementation.
- **Software Architecture.** This course provides the students with knowledge on how to design the architecture of software systems in a systematic way that addresses the required functionality and supports the necessary system qualities. The course provides students with a technical toolbox of concepts, methods, and principles to support the software design, implementation, and evaluation as much as a wider understanding of the context and domain of the system. The students will learn to analyze and adapt the software design approach to support iterative and agile development and to recover the architecture of existing systems by analyzing source code and related artifacts.
- **Practical concurrent and parallel programming.** In this course students learn how to write correct and efficient concurrent and parallel software, primarily using Java, on standard shared-memory multicore hardware such as laptops, desktops and servers. The course covers basic mechanisms such as threads, locks and shared memory as well as more advanced. It covers how to measure and understand performance and scalability of parallel programs. It covers tools and methods to find bugs in concurrent programs. The course places emphasis on how to reason about the correctness of concurrent programs.
- **DevOps, Software Evolution and Software Maintenance.** In this course, the students will discover all the software engineering activities that take place after an initial software product is delivered or after a legacy system is taken over from a theoretical and practical perspective. Students, through project work, will take over such a system that is live and serving users, they will refactor and migrate it to the languages and technologies of their liking. All subsequent DevOps, software evolution and maintenance activities will be performed directly on these systems.

Constituent elective courses:

The program offers a variety of constituent electives for the students to specialize within the overall goal of the program. Students are expected to select six electives 45 ECTS in total in the first four semesters. Minimum 30 ECTS – four courses – must be within these constituent electives. *The program does not constrain the combination of electives students can choose*, allowing students the flexibility needed to cater to their interests. These electives cover a range of topics across the following categories.

- **Software Development and Engineering:** Courses in this category deepen students' understanding of principles, techniques, and tools for designing, implementing, and evaluating modern software systems. Topics have an emphasis on building scalable, efficient, and robust systems.
- **Applied Data Science and Artificial Intelligence.** Courses in this category teach students to analyze, interpret, and utilize data effectively in data-driven software systems. Topics include data management, visualization, probabilistic modeling, and ethical considerations such as bias and fairness in machine learning. These courses prepare students to address challenges in developing real-world data-intensive systems.
- **Security and System.** Courses in this category focus on techniques addressing quality aspects such as the security and performance of software systems. Topics include fuzzing for security, risk management, IoT systems, and performance analysis. Students develop skills to secure and optimize computing infrastructures.
- **Business and Technology Management.** The courses in this category cover topics relevant to business strategy and organizational management. Taking these courses, the students gain skills to manage cross-functional teams, approach organizational challenges, and lead efforts in digital transformation.

Software Development and Engineering	Applied Data Science and Artificial Intelligence	Security and System	Business and Technology Management
• Advanced Software Engineering • Data-Driven Design & Development • Program Verification • Practical Parallel and Concurrent Programming • Advanced Programming • Software Ecosystems • Industrial Scrum Master Training	• Introduction to Artificial Intelligence • Big Data Processes • Geospatial Data Science • Probabilistic Programming • Algorithmic Fairness, Accountability and Ethics	• Ethical Hacking • Computer Systems Performance • Internet of Things • How to make (almost) anything • Applied information Security • Language based security • Data driven security	• Enterprise Architecture • Digital Organizing and Transformation • Reassembling Innovation • Managing Risk in IT and Cyber Security