

Addis Ababa Science and Technology University

School of Electrical Engineering and Computing

**Modularized Curriculum for B.Sc Degree in Software Engineering**

(Four Years Program)

Department of Computer Science & Information Technology

April 2015

# Background

These days, it is difficult to mention any are of business, education and research that didn’t utilize software. Software plays a central role in almost all aspects of daily life: in government, banking and finance, education, transportation, entertainment, medicine, agriculture, and law. The number, size, and application domains of computer programs have grown dramatically; as a result, hundreds of billions are being spent on software development, and the livelihood and lives of most people depend on the effectiveness of this development. Software products have helped us to be more efficient and productive. They make us more effective problem solvers, and they provide us with an environment for work and play that is often safer, more flexible, and less confining. For the above mentioned tasks, software engineering plays vital role. In addition, Software engineering is also a discipline, which involves the specification, design, construction and verification of large software systems.

Although the field of Software Engineering was established more than 30 years ago, Bachelor degree programs in Software Engineering in Ethiopia came into existence in the last few years. According to the GTP (Growth and Transformation Plan) goals and strategies set by the ministry of Science and Technology, the ministry emphasizes the fact that it will work extensively towards creating an enabling environment for ICT professionals to be engaged in innovation, creativity, and research activities. So that the ministry can effectively support GTP reform to ensure sustainable development, poverty eradication, human resource development and capacity development through well developed information communication technology involvement. Hence, the department of computer science and information technology in the school of Electrical Engineering and Computing has opened a new program in Software Engineering aiming to produce skilled graduates in the field of Software Engineering to satisfy the demand of manpower in the country.

The Software Engineering program has been started during the 2014-2015 academic year, with a target of admitting students in the regular program under the department of computer science and information technology in the school of Electrical Engineering and Computing. This degree program is a four year program designed for students who wish to pursue careers as professional software engineers in industry and commerce. Graduates of this program may pursue career opportunities in a variety of entry-level positions, including software developer, software programmer, application developer, Web application developer, software engineer, software applications analyst, Software Project Manager, Software Tester and Web applications analyst.

# Objective of the Program

The objectives of the Software Engineering curriculum are to enable students to:

* Be employed in industry, government, or entrepreneurial endeavors to demonstrate professional advancement through significant technical achievements and expanded leadership responsibility;
* Demonstrate the ability to work effectively as a team member and/or leader in an ever-changing professional environment; and
* Progress through advanced degree or certificate programs in software engineering, computing, science, engineering, business, and other professionally related fields.
* Assisting Government’s policy of increasing trained manpower, thereby reducing the shortage of skilled manpower that are required for different organizations, to develop the nation
* Assisting the industrial community and institutions around in the region to achieve excellence

# Graduate Profile

Graduates of the program are expected to demonstrate the ability to:

* Apply knowledge of mathematics, science, and engineering.
* Design and conduct experiments, as well as to analyze and interpret data.
* Design a system, component, or process to meet desired needs within realistic constraints such as economic, environmental, social, political, ethical, health and safety, manufacturability, and sustainability.
* Function on multi-disciplinary teams.
* Identify, formulate, and solve engineering problems.
* Understand professional and ethical responsibility.
* Organize and relate ideas in written, oral, and graphic form with coherence, clarity, and unity.
* Develop and present consistent arguments to selected and diverse audiences.
* Understand the impact of engineering solutions in a global, economic, environmental, and societal context.
* Engage in life-long learning.
* Acquire knowledge of contemporary issues.
* Use the techniques, skills, and modern engineering tools necessary for engineering practice.
* Analyze, design, verify, validate, implement, apply, and maintain software systems.
* Appropriately apply relevant topics in computer science and supporting disciplines to the design, development, evaluation and use of large and complex software systems.
* Work in one or more significant application domains.

**Knowledge and Understanding**

Software Engineering graduates must:

* Have strong analytical and critical thinking skills, which help them understand complex software and related systems.
* Demonstrate intellectual competency and articulate them effectively;
* Have in depth understanding of relevant theories, practices, languages and tools that may be deployed for the specification, design, implementation and evaluation of software systems.
* Capitalize such knowledge in modeling and design of automated systems.
* Design and implement software system solutions that enhance organizational performance by demonstrating skill in understanding and modeling organizational processes and data, defining and implementing technical and process solutions, managing projects, and integrating systems.
* Assist and participate in research

**Practical skill**

* Elicit user requirements;
* Identify problems and the critical information needed for software solution.
* Analyze and synthesize information and technologies, and draw reasoned inferences.
* Formulate and then evaluate appropriate problem-solving strategies.
* Model and Design large and complex software systems;
* Write application and system software;
* Manage software projects.

**Transferable skills**

* Ability to provide software engineering consultancy services
* Ability to apply entrepreneur skills

**Attitude and values**

* Software Engineering graduates must exhibit strong ethical principles, good interpersonal communication and team skills.

Software Engineering graduates must have environmental, social and cultural awareness.

# Academic Requirements

## 4.1 Admission Requirements

* The requirements for admission to the program are as per the Ministry of Education & Universities admission requirement.

## Duration of Study

* Under normal circumstances, four years of study are required to complete the program.

## Graduation Requirements

The major graduation requirements of the program include:

* Fulfilling the graduation requirements set by Addis Ababa Science and Technology University.
* Successful completion of a total of 140 credit points as per this curriculum;

## Medium of Instruction

The medium of instruction for the program is ENGLISH

# Course Coding

The course code will have four-digit number like SEabcd. The first digit, ‘a’, indicates the **year** of the course offering (‘1’ for 1st year, '2' for 2nd year, ‘3’ for 3rd year and ‘4’ for 4th year courses);the middle numbers, ‘bc’, indicate the **module** **number** in the program; the last number, ‘d’, indicates **order** of the course within the module.

# Assessment and Evaluation

The different abilities that graduates of this program must acquire are assessed as followed.

* Cognitive Abilities: Cognitive abilities are assessed by a combination of traditional written examinations and continuous assessment, including marked essays, class tests and computer programming problems.
* Practical Abilities: Practical Abilities are assessed mostly by continuous assessment. Some of practical abilities are assessed as part of group project module.
* Transferable skills: some skills, like the use of software tools and ability to communicate orally and in written form are directly assessed, in assignments or projects, other skills are not directly assessed but their effective use will enhance the students overall performance.
* Evaluation will be done based on the midterm exam, assignments, laboratory sessions demonstrations, Project work, and final exam and etc

## 7.1 Mode of Assessment

Mode of assessment is tobe treated as followed

For any course having lecture, laboratory/practice and tutorial, continuous assessment shall be used for each of them.

1. A student shall not score less than 50% in each lecture, laboratory/practice and tutorial assessments. If a student scores less than 50% , the instructor shall give additional assignment. However, if a student fails to score the pass mark according to the fixed scale, the instructor shall convert the grade into “F”.
2. For courses having lecture, laboratory and practicum, each will be evaluated out of 100%. The relative weight for each of these methods shall be proportional to their credit point. Such course shall be assessed by at least 3 assessments.
3. For courses having laboratory and practical activities, each will be evaluated out of 100%. The relative weight for each of these methods shall be proportional to their credit point. Such course shall be assessed by at least 3 assessments.
4. For courses having lecture, laboratory, practical and tutorial each will be evaluated out of 100%. The relative weight for each of these methods shall be proportional to their credit point. Such course shall be assessed by at least six assessments.
5. For a course to be delivered by lecture method only; 50% shall be assessed by continuous assessment constituting four assessments; while 50% shall be allotted for final exam.

## 7.2 Grading System

Grading is done as per the University Grading System. However; during Grading be sure that the following points are considered.

1. For any course having lecture, laboratory/practice and tutorial, raw mark evaluation should type of assessments.
2. The final number grade for a course is obtained based on the weighted average of lecture, laboratory/practice and tutorial.

## 7.3 Degree Nomenclature

The degree awarded to a student who has completed all the required courses of the program reads,

Bachelor of Science Degree in Software Engineering

***yúYNSÆClR Ä!G¶ bîFTê&Rx!NJn¶NG***

# Teaching-Learning Methods and Materials

The basic teaching method is lecture supported by tutorial, design projects, practical laboratory exercises and, in some cases, by simulation. E-learning and audio-visual supports are also used to aid lectures. Students are assigned to work on projects to enable them enhance innovative and independent working ability. The industry internship is also an integral part of the teaching/learning process. Students are provided with handouts and teaching materials.

1. **List of Modules**

***Module 01: Software Engineering Basics***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr. | Lec | Lab | Tut | H/S |
| SE 1012 | Introduction to Software Engineering and Computing | Compulsory | 7 | 4 | 0 | 0 | 0 | 5 |
| EnLa202 | Basic Writing Skills | Compulsory | 5 | 3 | 3 |  | 0 | 6 |
| EnLa201 | Communicative English | Compulsory | 5 | 3 | 3 |  | 0 | 6 |
| CvEt1201 | Civics and Ethics | Compulsory | 5 | 3 | 2 | 0 | 0 | 4 |
| Math 264 | Calculus I | Compulsory | 7 | 4 | 4 | 0 | 2 | 4 |
| EEng 1202 | Digital Logic Design | Compulsory | 5 | 3 | 2 | 3 | 0 | 5 |
| Econ 101 | Introduction to Economics | Compulsory | 5 | 3 | 3 | 0 | 0 | 3 |
| **Total** | | | **33** | **19** |  |  |  |  |

***Module 02: Basic programming***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr. | Lec | Lab | Tut | H/S |
| Math 192 | Discrete Mathematics I | Compulsory | 5 | 3 | 3 | 0 | 2 | 7 |
| SE 1021 | Programming Fundamentals I | Compulsory | 5 | 3 | 2 | 2 | 1 | 5 |
| SE 1022 | Programming Fundamentals II | Compulsory | 5 | 3 | 2 | 2 | 1 | 5 |
| SE 1023 | Computer Organization and Architecture | Compulsory | 7 | 4 | 3 | 3 | 0 | 7 |
| **Total** | | | **22** | **13** |  |  |  |  |

***Module 03: Advanced Programming***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr | Lec | Lab | Tut | H/S |
| SE 2031 | Object Oriented Programming | Compulsory | 7 | 4 | 3 | 3 | 0 | 5 |
| Math 292 | Discrete Mathematics II | Compulsory | 5 | 3 | 3 | 0 | 2 | 7 |
| SE 2032 | Data structures and Algorithms | Compulsory | 5 | 3 | 2 | 2 | 1 | 5 | |
| SE 2033 | Advanced programming in java | Compulsory | 7 | `4 | 3 | 3 | 0 | 7 |
| SE 2034 | Mobile Programming | Compulsory | 5 | 3 | 3 | 3 | 0 | 5 |
| **Total** | | | **29** | **17** |  |  |  |  |

**Module 04: System Software Development**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr | Lec | Lab | Tut | H/S |
| SENG2041 | Formal Language and Automata Theory | Compulsory | 7 | 4 | 4 |  | 0 | 5 |
| SENG3042 | Principles of Compiler Design | Compulsory | 7 | 4 | 3 | 3 | 0 | 5 |
| SENG2043 | Operating Systems | Compulsory | 7 | 4 | 3 | 3 | 0 | 5 |
| SENG2044 | System Programming | Compulsory | 5 | 3 | 3 | 3 | 0 | 5 |
| **Total** | | | **26** | **15** |  |  |  |  |

***Module 05: Computer Networking***

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr | | Lec | Lab | Tut | H/S |
| SE 3051 | Data Communication and Computer Networks | Compulsory | 7 | | 4 | 3 | 3 | 0 | 5 |
| SE 4052 | Information Security | Compulsory | 5 | | 3 | 2 | 3 | 0 | 5 |
| **Total** | | | **12** | **7** | |  |  |  |  |

***Module 06: Database Systems***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr | Lec | Lab | Tut | H/S |
| SE 2061 | Database Management Systems | Compulsory | 7 | 4 | 3 | 3 | 0 | 5 |
| SE 3062 | Distributed Systems | Compulsory | 5 | 3 | 2 | 3 | 0 | 7 |
| SE 3063 | Data mining and data ware housing Elective | Compulsory | 5 | 3 | 2 | 3 | 0 | 7 |
| **Total** | | | **12** | **7** |  |  |  |  |

***Module 7 : Web Technology***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr | Lec | Lab | Tut | H/S |
| SE 3071 | Web Design and Programming | Compulsory | 7 | 4 | 2 | 3 | 0 | 5 |
| SE 2072 | WEB Service | Compulsory | 5 | 3 | 2 | 3 | 0 | 5 |
| SE 3083 | Multimedia System | Compulsory | 5 | 3 | 2 | 3 | 0 | 5 |
| **Total** | | | **12** | **7** |  |  |  |  |

***Module 08: Software Requirement Modeling and Analysis***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr. | Lec | Lab | Tut | H/S |
| SE 2081 | Object Oriented System analysis and modeling | Compulsory | 5 | 3 | 2 | 3 | 0 | 5 |
| SE 3082 | Software Requirements Engineering | Compulsory | 5 | 3 | 3 | 0 | 0 | 6 |
| **Total** | | | **10** | **6** |  |  |  |  |

***Module 09: Software Design***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr. | Lec | Lab | Tut | H/S |
| SE 3091 | Software Design and Architecture | Compulsory | 5 | 3 | 3 | 0 | 0 | 7 |
| SE 3092 | Software Engineering approaches to Human Computer Interaction | Compulsory | 5 | 3 | 2 | 3 | 0 | 7 |
| SE 3095 | Software Component Design | Compulsory | 5 | 3 | 2 | 3 | 0 | 5 |
| SE 3096 | Software engineering tools and practices | Compulsory | 5 | 3 | 2 | 3 | 0 | 5 |
| **Total** | | | **15** | **12** |  |  |  |  |

***Module 10 System Modeling and simulation***

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | | Cr | Lec | Lab | | Tut | H/S |
| SE 3101 | Computer Graphics | Compulsory | 5 | | 3 | 2 | 3 | | 0 | 7 |
| SE 3102 | Multimedia System | Compulsory | 5 | | 3 | 2 | 3 | | 0 | 7 |
| Stat 173 | Probability and Statistics | Compulsory | 5 | | 3 | 3 | 0 | | 2 | 5 |
| SE 3103 | Simulation and Modeling | Compulsory | | 5 | 3 | 2 | 3 | 0 | | 5 |
| **Total** | | | **20** | | **12** |  |  |  | |  |

***Module 11: Software Management***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| SE 4112 | Software Project Management | Status | 5 | 3 | 3 | 0 | 0 | 7 |
| MGMT3161 | Entrepreneurship | Compulsory | 5 | 3 | 3 | 0 | 0 | 3 |
| SE 4113 | Software Quality Assurance and Testing | Compulsory | 5 | 3 | 3 | 0 | 0 | 7 |
| SE 4111 | Process management | Compulsory | 5 | 3 | 2 | 3 | 0 | 6 |
| SE 4119 | Practical Attachment | Compulsory | 3 | 2 | 2 | 0 | 0 | 5 |
| **Total** | | | **25** | **15** |  |  |  |  |

***Module 12: Research and Project Works***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Course No | Title | Status | ECTS | Cr. | Lec | Lab | Tut | H/S |
| SE 4122 | Seminar in Software Engineering | Compulsory | 2 | 1 | 0 | 0 | 3 | 5 |
| SE 4123 | Software Engineering Capstone project I | Compulsory | 7 | 4 | 0 | 0 | 6 | 9 |
| SE 4124 | Software Engineering Capstone project II | Compulsory | 7 | 4 | 0 | 6 | 0 | 9 |
| SE 4121 | Research methodology in Software Engineering | Compulsory | 3 | 2 | 2 | 0 | 0 | 5 |
| SE 4122 | Selected topics in Software Engineering | Compulsory | 2 | 1 | 0 | 0 | 3 | 5 |
| **Total** | | | **19** | **11** |  |  |  |  |

1. **Course/Module Breakdown**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year I Semester I** | | | | | | |
| **Module No** | **Course Code** | **Course Title** | **ECTS** | **Cr.Hr** | **No of weeks** | **Delivery** |
| 01 | EnLa 201 | Communicative English | 5 | 3 | 16 | Parallel |
| Math 264 | Calculus I | 7 | 4 | 16 | Parallel |
| SE 1011 | Introduction to Computing and Software Engineering | 7 | 4 | 16 | Parallel |
| PHY206 | Electricity and Magnetism | 5 | 3 | 16 | parallel |
| 02 | SE 1021 | Fundamentals of Programming I | 7 | 4 | 16 | Parallel |
|  |  | **Total** | **31** | **18** |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year I Semester II** | | | | | | |
| **Module No** | **Course Code** | **Course Title** | **ECTS** | **Cr.Hr** | **No of Weeks** | **Delivery** |
| 01 | CvEt 201 | Civics and Ethics | 5 | 3 | 16 | Parallel |
| EnLa.202 | Basic writing skills | 5 | 3 | 16 | Parallel |
| 02 | SE 1022 | Fundamentals of Programming II | 7 | 4 | 16 | Parallel |
| Math 192 | Discrete Mathematics I | 5 | 3 | 16 | Parallel |
| SE 1023 | Digital logic design | 5 | 3 | 16 | Parallel |
| 01 | Econ 101 | Introduction to Economics | 5 | 3 | 16 | Parallel |
|  |  | Total | 32 | **19** | 16 | Parallel |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year II Semester I** | | | | | | |
| **Module No** | **Course No** | **Title** | **ECTS** | **Cr.Hr** | **No of Weeks** | **Delivery** |
| 03 | SE 2031 | Object Oriented Programming | 7 | 4 | 16 | Parallel |
| SE 2032 | Data structures and Algorithms | 5 | 3 | 16 | Parallel |
| Math 292 | Discrete Mathematics II | 5 | 3 | 16 | Parallel |
| 04 | SE 2043 | Computer Organization and Architecture | 7 | 4 | 16 | Parallel | |
| SE 2041 | Fundamentals of Database Systems | 7 | 4 |  |  |
|  |  | **Total** | **31** | **18** |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year II Semester II** | | | | | | |
| **Module No** | **Course No** | **Title** | **ECTS** | **Cr.Hr** | **No of Weeks** | **Delivery** |
| 03 | SE 2033 | Advanced programming | 7 | 4 | 16 | Parallel |
| SE 2031 | Data Communication and Computer Networks | 7 | **4** | 16 | Parallel |
| 04 | SE 2041 | Formal Language and Automata Theory | 7 | 4 | 16 | Parallel |
| Stat 173 | Probability and Statistics | 5 | 3 | 16 | Parallel |
| 06 | SE 2061 | Web Design and Programming | 7 | 4 | 16 | Parallel |
|  |  | Total | 29 | **19** |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year III Semester I** | | | | | | |
| **Module No** | **Course Code** | **Course Title** | **ECTS** | **Cr.Hr** | **No of Weeks** | **Delivery** |
| 08 | SE 3081 | Object Oriented Software Requirement Analysis and modeling | 5 | 3 | 16 | Parallel |
| 06 | SE 3062 | Distributed Systems | 5 | 3 | 16 | Parallel |
| 05 | SE3054 | Systems Programming | 7 | 4 | 16 | Parallel |
| 07 | SE3072 | Principles of Compiler Design | 7 | 4 | 16 | Parallel |
|  | SE 2073 | Operating Systems | 7 | **4** | 16 |  |
|  |  | **Total** | **29** | **18** | 16 | Parallel |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Year III Semester II** | | | | | | |
| **Module No** | **Course Code** | **Course Title** | **ECTS** | **Cr.Hr** | **No of Weeks** | **Delivery** |
| 07 | SE 3071 | Software Project Management | 5 | 3 | 16 | Parallel |
| SE 3074 | Human Computer Interaction | 5 | 3 | 16 | Parallel |
| 08 | SE 3082 | Software Requirements Engineering | 5 | 3 | 16 |  |
| 03 | SE 3034 | Wireless Communications & Mobile Computing | 5 | 3 | 16 | Parallel |
| 10 | SE 3101 | Computer Graphics | 5 | 3 | 16 |  |
| 09 | SE 3091 | Software Design and Architecture | 5 | 3 | 16 | Parallel |
|  |  | **Total** | **30** | **18** |  |  |

-

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Year IV Semester I** | | | | | |
| **Module No** | **Course Code** | **Course Title** | **ECTS** | **Cr.Hr** | **No of Weeks** | **Delivery** |
| 09 | SE 4091 | Software Engineering Tools and Practices | 5 | 3 | 2 | Parallel |
| 12 | SE 4123 | Software Engineering Capstone project I | 7 | 4 | 16 | Parallel |
| SE 4121 | Research Methods in Software Eng. | 3 | 2 |  |  |
| SE 4122 | Seminar in Software Engineering | 3 | 1 | 16 | Parallel |
| 11 | SE 4119 | Practical Attachment | 3 | 1 |  |  |
| 09 | SE 3095 | Software Component Design | 5 | 3 |  |  |
| 10 | SE 3104 | Simulation and Modeling | 5 | 3 | 16 | Parallel |
|  |  | **Total** | **28** | **17** |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Year IV Semester II** | | | | | |
| **Module No** | **Course Code** | **Course Title** | **ECTS** | **Cr.** | **No of Weeks** | **Delivery** |
| 09 | SE 4052 | Computer System Security and Information System | 5 | 3 | 16 | Parallel |
| SE 4113 | Software Quality Assurance and Testing | 5 | 3 |  |  |
| MGMT3161 | Entrepreneurship | 3 | 3 | 16 | Parallel |
| 07 | SE 3073 | Multimedia System | 5 | 3 | 16 |  |
| 12 | SE 4123 | Software Engineering Capstone Project II | 7 | 4 | 16 | Parallel |
| SE 4122 | Selected topics in Software Engineering | 2 | 1 |  |  |
|  |  | **Total** | **27** | **17** |  | Parallel |

**11. Course Description**

**Year I**

**Course: Introduction to Computing and Software Engineering**

**Course description**

This course is intended to equip students with theoretical and practical aspects of fundamentals of Computer Science and software engineering by furnishing them with a broad oversight of the discipline of formal computer science and software engineering:

* Introduces students to the definition of computer science, software engineering, algorithms and computer science as a discipline
* significantly Introduces students to the computer system
* Introduces students to data representation in computer system
* introduces and provides an insight to the basic science of computer architecture
* Introduces students to data communication and computer networks
* Introduces students to potential security threats to computers and methods to safeguard computer system.

**Course: Fundamental of Programming I**

**Course description**

The course aims to help students gain experience and confidence with concepts of computer programming that will allow them to solve meaningful problems in a variety of fields. Basic ideas of computer and software architecture will be covered along with programming concepts such as flow, logic, data structures. The core emphasis of the course is on the following: 1) Programming Concepts – recognizing and understanding the fundamental constructs present in all programming languages; 2) Design Methodology – Using abstraction to simplify complex problems into concrete subtasks; 3) Algorithmic Thinking – Defining the process used to glean information from data, and being able to convince oneself of its correctness; 4) Literate Programming – Writing programs which are clearly structured and can be read by a human as well as a machine; 5) Methodical and efficient development of computer programs using step-wise refinement and incremental testing and debugging.

**Course: Fundamental of Programming II**

**Course description**

This course is a continuation of Fundamentals of Programming I. It deals about modularity, dynamic memory management, pointers and file system. That is, the course focuses on topics which shall further enrich students' knowledge and skill in structured programming.

**Course: Digital Logic Design**

**Course description**

This course is a study of digital circuit fundamentals with an emphasis on combinational and sequential logic design, Boolean algebra and switching theory, logic simplification and implementation using standard digital IC's of various logic families and programmable logic devices. It provides a modern introduction to logic design and the basic building blocks used in digital systems, in particular digital computers. It discusses about combinational logic: logic gates, minimization techniques, arithmetic circuits, and modern logic devices such as field programmable logic gates. It also deals with sequential circuits: flip-flops, synthesis of sequential circuits. State machines will then be discussed and illustrated through case studies of more complex systems using programmable logic devices. Different representations including truth table, logic gate, timing diagram, switch representation, and state diagram will be discussed.

**Year II**

# Object Oriented Programming (SE 2031)

1. **Course Description**:

This programming course emphasizes the methodology of programming from an object-oriented perspective and software engineering principles. It allows students to develop the ability to analyses programming problems and design and document suitable solutions and to implement reliable and robust software using contemporary program design methods. Topics to be dealt with are: classes: data abstraction, information hiding, overloading; inheritance; polymorphism; exceptions handling.

1. **Course Goals or Learning Outcome:**

Upon successful completion of the course, students will be able to:

* Explain the basic object oriented concepts
* Successfully code, debug and run programs with appropriate development environment
* Work on more advanced programs written in Java
* Have clear differentiation between structural and object oriented programming paradigms

# Data structures and Algorithms (SE 2032)

1. **Course Description**:

This course focuses on the study of data structures, algorithms and program efficiency. Topics include: analysis of time and space requirements of algorithms; program efficiency improving techniques, abstract data types such as linked lists, stacks, queues, trees (traversal, implementations); simple searching algorithms (linear search, binary search, … ), simple sorting algorithms (bubble sort, insertion sort, selection sort, … ), advanced sorting algorithms (merge sort, quick sort, heap sort … )

1. **Course Goals or Learning Outcome:**

The course aims:

* To introduce the most common data structures like stack, queue, linked list
* To give alternate methods of data organization and representation

To enable students use the concepts related to Data Structures and Algorithms to solve real world problems

* To practice Recursion, Sorting, and Searching on the different data structures
* To implement the data structures with a chosen programming language

# Computer Organization and Architecture (SE 1023)

1. **Course Description**:

An overview of Computer Science; historical development of computers; logical organization of a computer system; software ,data representation inside computers; computer arithmetic; computer system architecture(Boolean algebra and logic circuits); overview of computer networks & Computer security, Lab Introducing the use of PC in an MS DOS, MS Windows environment; Introduction to main MS application packages in that context – MS Excel, MS Word, MS access, MS power point, Introduction to using Internet;

1. **Course Goals or Learning Outcome:**

Up on the completion of this course, students will be able to:

* Find out what computer science is about and explore its applications and impact in other disciplines
* Explain how the computer understands words, numbers, and how data and instructions are stored, retrieved and processed
* Describe basic characteristics of components of a computer system, Illustrate and analyze basics of Computer architecture
* Demonstrate basics of networking. explain what the Internet is , browse and use the Internet for information and communication purpose ,
* Discuss and explain importance of securing computer-based resources (including hardware, software, and information)

# Fundamentals of Database Systems (SE 2061)

1. **Course Description**:

This course is assumed to be the first course in database management systems. It starts by introducing what database is and database systems, and how it differs with the traditional file processing system. It then deals with data models, ER diagrams, database design methods, normalization, relational algebra and calculus, file organizations and storage management, index structure for files, and SQL statements. The course mainly emphasizes on basics of database systems how to retrieve and modify data. It also deals different database environments.

1. **Course Goals or Learning Outcomes:**

By the end of this course, students will be able to:

* Understand what database is, database system and DBMS
* Differentiate database system from file system
* Identify the pros and cons of manual approach, file based approach and database approach
* Understand the basic principles of database design systems using different database models
* Appreciate the use of database system in the real world.
* Design different types of databases
* Understand database normalization & functional dependency
* Understand the principles of relational database management systems and their languages
* Understand file organizations and storage management, and index structure for files
* Demonstrate queries in the relational algebra.
* Demonstrate queries in the tuple relational calculus.
* Create a relational database schema in SQL that incorporates key, entity integrity, and referential integrity constraints.

# Advanced programming (SE 2033)

1. **Course Description**:

This course covers topics on Java which includes:, AWT and Swing, I/O Streams, Multithreading, Network Programming, Java database connectivity (JDBC), RMI, aand Introduction to Servlets.

1. **Course Goals or Learning Outcome:**

At the end of this course, students should be able to:

* Carry out design and development of complex elements, such as user interfaces, multiprocessing, and fault-tolerant components;
* Write TCP/IP Client Server applications using Sockets;
* Write Java applications using the JDBC to make database independent queries; and
* Call methods remotely.

# Data Communication and Computer Networks (SE 3051)

1. **Course Description**:

This course will explore the various types of the data communication systems, networks and their applications. Concept & terminologies like computer networks, layer architecture (OSI & TCP/IP), network hardware, network software, standardization, network medium, and IP addressing will be explored. The practical aspect will deal with building small to medium level networks including Cabling, Configuring TCP/IP, Peer to Peer Networking, Sharing resources, Client Server Networking.

1. **Course Goals or Learning Outcomes:**

By the end of this course, students will be able to:

* Understand the concepts and principles of data communications and computer networks
* Understand data transmission and transmission media
* Understand Protocols and various networking components
* Understand TCP/IP & OSI Reference Model
* UnderstandLAN and WAN technologies
* Understand and implement IP addressing.
* Build small to medium level Computer networks

# Formal Language and Automata Theory (SENG 2041)

1. **Course Description**:

This course aims to develop the theoretical foundations of computer science through study of mathematical and abstract models of computers and the theory of formal languages. It also, introduces some fundamental concepts in automata theory and formal languages including grammar, finite automaton, regular expression, formal language, pushdown automaton, and Turing machine. Not only do they form basic models of computation, they are also the foundation of many branches of computer science, e.g. compilers, software engineering, concurrent systems, etc.

1. **Course Goals or Learning Outcomes:**

On completion of this course students should be able to:

* Introduce concepts in automata theory and theory of computation
* Study the central concepts of automata theory
* Acquire insights into the relationship among formal languages, formal grammars, and automata.
* Identify different formal language classes and their relationships
* Design grammars and recognizers for different formal languages
* Prove or disprove theorems in automata theory using its properties
* Familiar with thinking analytically and intuitively for problem solving situations in related areas of theory in computer science.

# Web Design and Programming (SE 3071)

1. **Course Description**:

Client-server Architecture: Web page Design and development; information Architecture & visualization; static & dynamic pages, client side programming using scripting languages (JavaScript), OOP programming using PHP, File manipulation and Management using PHP. Multimedia web programming, dynamic web content creation; session tracking; Internet protocols, PHP Web controls, connecting web page to data base, loading and manipulating XML documents, web security; cryptography theory.

1. **Course Goals or Learning Outcomes:**

By the end of this course, students will be able to:

The objective of this module is to familiarize students: the concepts, principles and methods in programming for web and Internet environment and the services and technologies available and used on Internet.

Specifically, at the completion of this module the student is able to

* Provide fundamental concepts and skills for the understanding and development of web-based applications.
* Construct Web sites using HTML and Server Side scripting Languages.
* Implement client-side interactivity
* use CSS to manage Web site presentation and maintenance
* Select and customize existing technologies to set up and maintain web servers
* Specify, build and manage form and content of information-rich web sites
* Design, implement and evaluate client-server systems following specific protocol specifications, taking into account concurrency issue

**Year III Semester I**

1. **Object Oriented Software Requirement Analysis and modeling (SE 3081)**

This course provides concepts on how to capture software requirements and handle difficult situations in gathering data to build systems. Special emphasis is given to working with clients and to learning about the needs of users who interact with a system. The course addresses elicitation, specification, and management of software system requirements.

The primary function of an analyst in the requirements capture process is to analyze and organize informal requirement statements into a form that can be verified by a user and used as input to design. Requirements specifications must also provide the sole criteria for the validation of the end software product. In achieving these goals an effective analyst must bring to a project the ability to listen to users’ needs and create a complete, correct, consistent and unambiguous requirement specification. Analysis models describing functional and time dependent behavior and data relationships are therefore an essential component of the analyst's tool box. Useful models also act as an aid to understanding complex systems, representing the essence of a system in graphical form and downplaying excessive detail.

1. **Distributed Systems (SE 3062)**

A distributed system is a computer system consisting of several independent computers, connected by a network, that can work together to perform a task or provide a service. These systems run special software that allows for transparent sharing of the distributed computing resources and data. Typical examples include: the World Wide Web, cloud computing, networked file systems, DNS, and massive multiprocessor supercomputers.

This course aim to provide students with a deeper understanding of distributed systems. In particular focus on the principles, techniques, and practices relevant to the design and implementation of such systems. The course takes a systems-oriented view of distributed systems, concentrating on infrastructure software and providing hands-on experience implementing distributed systems.

1. **Systems Programming (SE2044)**

This course introduces computer systems with a basic understanding of the issues involved in writing system programs, manipulating system processes, system I/O, system permissions, files, directories, signals, threads, sockets, terminal, etc., including interaction of hardware and software through the operating system, from the programmer's perspective. The fundamental abstractions are emphasized; to provide programmers a common interface to a wide variety of hardware devices. Topics covered include linking, system level I/O, concurrent programming, and network programming.

1. **Principles of Compiler Design (SE3042)**

This course introduces fundamental concepts in the design and implementation of compilers. It includes basic concepts, compiler components, lexical analysis, syntax analysis, symbol tables handling, parsing techniques, error handling and recovery, syntax-directed translation, type checking, run-time organization, intermediate code generation, code generation and optimization.

1. **Operating Systems (SE 2043)**

This course examines the important concepts in operating system design and implementation. The operating system provides an established, convenient, and efficient interface between user programs and the bare hardware of the computer on which they run. The operating system is responsible for sharing resources (e.g., disks, networks, and processors), providing common services needed by many different programs (e.g., file service, the ability to start or stop processes, and access to the printer), and protecting individual programs from interfering with one another. Particular emphasis will be given to major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management (segmentation, paging, swapping), and file systems.

**Year III Semester II**

1. **Software Project Management(SE 4091)**

This course presents topics, techniques and practices critical to the management of software product development, including estimating, planning, quality metrics and assurance, tracking, team organization, and people management. It gives special attention to problems unique to software processes and development.

The course includes issues to select or adapt a lifecycle/model, some difficulties in communication and coordination, ethical and legal issues related to software projects.

1. **Human Computer Interaction (SE 3074)**

This course provides an introduction to and overview of the field of human-computer interaction (HCI). HCI is an interdisciplinary field that integrates theories and methodologies from computer science, cognitive psychology, design, and many other areas. The course will span current theory and practice in interface specification, design, implementation and evaluation of a user interface. The students will be incrementally led through the phases of ethnographic study and requirements analysis, scenario-based design, paper prototyping, computer prototyping, and several methods of usability analysis and evaluation. The course involves also an introduction to software architectures used in modern graphical user interfaces, including the implementation of a few simple interfaces using different toolkit.

1. **Software Requirements Engineering (SE 3082)**

This course provides a broad overview of all aspects of requirements engineering focusing on paradigms, methodologies, techniques and tool. The course has a strong practical element, both in terms of the introduction to the topic, and systematically establishing, defining and managing the requirements for large, complex, changing and software-intensive systems.

This course explores not just why the requirements matter, but how to gather the right requirements, document them effectively and ensure that they are properly implemented. It explores the roles of the requirements engineer and business analyst both in hunting out the requirements and interacting with the project team to ensure the customer gets the product they want and need.

In this course you will earn effective techniques to select the right people from whom you will elicit the requirements. You will learn to document and track these requirements so nothing gets missed. You will learn which team members you must interact with at which stage in the project, and you will learn to track the requirements to completion.

1. **Wireless Communications & Mobile Computing (SE 2034)**

This course enables the student to learn the newest and fastest developing fields in the discipline and acquire the knowledge related to Wireless Communications and Mobile Computing. This includes the Principles, Components, Technologies, Architectures, and Modes of Wireless communications and Mobile Computing.

1. **Computer Graphics (SE 3101)**

Computer Graphics is a study of the hardware and software principles of interactive raster graphics. This course is to provide a broad and comprehensive exposure to the computer graphics field. Topics include an introduction to the basic concepts and image processing, geometric transformations, geometric modeling of curves and surfaces, animation, 2-D and 3-D modeling and transformations, viewing transformations, projections, rendering techniques, visibility algorithms, graphical software packages and graphics systems. Students will use a standard computer graphics API to reinforce concepts and study fundamental computer graphics algorithms.

1. **Software Design and Architecture (SE 3091)**

This course covers the fundamental design principles and strategies for software architecture and design, including the structural view of software components, and their characteristics and interrelationships, at a high level of abstraction. The course also covers the design principles that govern the purpose, structure, development, and evolution of software components.

Topics included are architectural styles, quality attributes notations and documents, reference architecture, domain specific architecture in architecture process and pattern-oriented design, component-oriented design, aspect-oriented design, and interface design. The course uses software design tools to reinforce design processes and representations.

|  |  |
| --- | --- |
| **Year IV** | |
| **Course Title** | **Course Description** |
| Software Engineering Tools and Practices | This course provides an introduction to the software engineering tools and practices - a look at typical approaches software engineers use to create applications in practice. Topics include requirements analysis, high-level design, detail-level design, UML modeling, code generation, basic unit testing, application building, continuous integration, and revision management. Laboratory assignments provide an opportunity for students to develop an understanding of these tools and how they are used in actual practice.  It also assesses current software engineering tools and practices. Topics include software architecture, integrated development environments, source code control systems, build and make systems, debuggers, static analysis tools, dynamic analysis tools, design tools, program verification, and design patterns. Experiences in multiple environments (Windows and a UNIX-based environment, i.e., MAC OS, Linux, or Solaris) |
| Software Engineering Capstone project I | Emphasizes teamwork in small groups on a substantial project that will be performed for a real customer. Projects are chosen so as to provide interdisciplinary content with project proposals being solicited from the community at large. Projects that involve students as well as customers from other disciplines are encouraged. Lectures will be directed toward the management of software development projects such as those being carried out by the teams. It is the intent of the course to provide a capstone experience that integrates the materials contained in the remainder of the software engineering or computer science curriculum through work on a project that applies this material in another discipline. |
| Research Methods in Software Eng. | This course concerns us with the topic of research. It attempts to define what research is, why we do research, and the various methods that researchers use to investigate problems.  It is designed as an under-graduate introduction to research methodology. Students are assumed to be familiar with statistics (such as mean, standard deviation, percentiles, and other such terms). The course provides a framework for conceptualizing research and is meant to underpin the research project for the final year as well as for the projects delivered for each subject.  The course will explore the role of empiricism in software engineering (SE) research, and will prepare students for advanced research in SE by examining how to plan, conduct and report on empirical investigations. The course will cover all of the principal methods applicable to SE: controlled experiment, case studies, surveys, archival analysis, action research and ethnographies, and will relate these methods to the relevant application of science. The course will critically review published examples of work that used each of the principle methods, both from within SE and from other disciplines. The course will cover techniques applicable to each of the steps of a research project, including formulating research questions, theory building, data analysis (using both qualitative and quantitative methods), building evidence, assessing validity, and publishing |
| Seminar in Software Engineering | This course investigates some topics (preferably advanced ones) in software engineering. Typically every time chooses a different theme. During the course, the student is expected to research a topic of his/her choosing that fits the theme, find papers and books, and report about the findings in form of presentations. A final report needs to be submitted which summarizes the findings. The seminar can be conducted as block course and consists of the presentations of the participants' contributions.  It can be held either in the fashion of scientific workshop or conference. Therefore, it will be based on the following principle:   * The participants choose the topic of their assignment themselves (proposals will be available). They describe the planned contribution in a short abstract which is then handed in, reviewed, and accepted or modified to meet the goals of the workshop. * Based on the abstract a research paper is written that will be reviewed by the participants and the supervisors in order to give feedback. The review will be a "blind review", i.e. neither the author nor the reviewer will be known to each other. |
| Practical Attachment | Field attachment will be implemented through partnership between the University and other partners who consume the University’s products and services. This partnership has three key stakeholders namely: the university, the students and the university partners. Commitment to this partnership rests on mutual benefits for all the key partners.  The expected benefits of the program for the different parties include:  **The University:**   * Will get opportunity to appreciate client demands and the quality of graduates required to fulfill these demands. * As  a  result  of  the  cooperation  the  potential  for  research  will  be enhanced and developed  with the various partners. * Will get opportunity to access training facilities and resources that are not available at AASTU.   **Students:**   * Will acquire new knowledge, skills, attitudes and practical experiences. * Will improve their confidence in problem solving. * Opportunity to relate to different categories of people likely to be met in the real-life situation. * Will gain exposure to the demands and challenges of the work place. * Will get opportunity to meet and work with potential employers. * Will have improved appreciation of the profession and therefore better work ethics.   **University Partners:**  Will get opportunity to be involved in the training and review of programs at AASTU.   * Will be exposed to a pool of potential employees from which to select. This will ultimately reduce the cost of induction/orientation of new employees. * Will get additional well trained human resource for effective and efficient service delivery. |
| Software Component Design | This course covers a wide range of component-based software development skills, from analyzing and modeling a problem with component-based notations and architectures, to implementing a solution using a particular component technology. The principles and methodologies in component based software development will be discussed in depth focusing on component-oriented programming and its related technologies. Component-based tools and languages, approaches for implementation of component-based software, including designing, building, assembling, and deploying reusable and COTS and in-house software components are discussed. Students will do projects focused on the life cycle of software components in various component technologies.  The course will review basic object-oriented techniques and how they support the creation of component based designs. Will Discuss an in-depth study of the use of software components in software architecture, design, implementation and testing. To discuss the issues involved in component re-use, including selection, adaptation and integration. |
| Simulation and Modeling | The course covers modeling and simulation concepts and discrete-event in particular. Simulation is the process of designing a model of a system and conducting experiments to understand the behavior of the system and/or evaluate various strategies for the operation of the system. Modeling & Simulation (M&S) has become an important tool in all phases of the acquisition process and can be used within most systems life cycle processes, including requirements analysis, architectural design, design & development, test and verification and operations and maintenance.  Application of theories, methods, and practices are covered during the semester. The course materials are divided into three parts. The first part provides background review and discussion on systems modeling concepts and overview of object-oriented programming languages. The second part provides background review and discussion on systems modeling concepts and overview of object-oriented programming languages. This part contains comprehensive descriptions on how to formulate and execute (simulate) models in a software engineering-like lifecycle. Students are engaged in detailed study of modeling elements, simulation protocols, and their relationships including experimentation. In-class description of modeling and simulation techniques will be illustrated through examples. During the semester students will gain  hands-on experience (via homework assignments and project). Students will create increasingly more complex models, which can be subsequently simulated and analyzed. The third part focuses on selected advanced topics aiding individual and team members’  projects. An important part of the course experience is through the class project. |