MAKERERE UNIVERSITY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

DEPARTMENT OF NETWORKS

P.O. BOX 7062, KAMPALA, UGANDA

PROPOSED BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

(B.Sc. SE) DEGREE PROGRAMME

June 2009

(DAY/ EVENING PROGRAMME)

Contents

1 Introduction

1.1 Background

1.1.1 Faculty of Computing and Information Technolog

The Faculty of Computing and Information Technology (CIT) was established by the University Council on 19th January 2005 by upgrading the Institute of Computer Science into a faculty with four departments of computer science, networks, information technology and information systems. The Institute of Computer Science which was established by the University Council in 1985, grew out of the University Computer Centre. CIT is mandated to run programmes in Computing, Information Technology and related areas. **Computing** and Information Technology.

1.1.2 Department of Networks

The department of Networks is currently responsible for running programmes in the following areas: Data Communications Engineering; Software Engineering; Hardware Engineering; Network and System Administration; Network Planning, Design and Management; Data Communication Networks; ICT Policy and Regulation; and related areas. The Department currently runs the following programmes approved by University Council:

1. PhD in Software Engineering;
2. M.Sc. in Data Communications and Software Engineering (Data Communications Engineering option; Network and System Administration option; Software Engineering option);
3. Postgraduate Diploma in Data Communications and Software Engineering (Data Communications Engineering option; Network and System Administration option; Software Engineering option);
4. Postgraduate Diploma in ICT Policy and Regulation

In addition the department runs the following professional (short) courses:

1. Cisco Certified Network Associate (CCNA);
2. Cisco Certified Network Professional (CCNP);
3. Microsoft Certified Systems Engineer (MCSE).

The Department of Networks does not have a bachelor's degree programme whose graduates would feed into the already established graduate programmes thus the proposed B.Sc. in Software Engineering.

1.1.3 Computing

Computing is concerned with the understanding, design and exploitation of computation and computer technology. It is a discipline that blends elegant theories (including those derived from a range of other disciplines such as mathematics, engineering, psychology, graphical design or well-founded experimental insight) with the solution of immediate practical problems; it combines the ethos of the scholar with that of the professional; it underpins the development of both small scale and large systems that support organizational goals. A degree programme or a programme component in the case of joint degree, will count as lying within the area of computing if the existence of computers and associated technology is seen as a central driving force in its motivation.

The following headings give a high-level characterisation of the whole area of computing, based on traditional hardware/ software and theory/ practice spectra, and including communication and interaction which spans across these areas:

1. Hardware:

- Computer architecture and construction
- Processor architecture
- Device-level issues and fabrication technology

2. Software:
   - Programming languages
   - Software tools and packages
   - Computer applications
   - Structuring of data and information
3. Communication and interaction:
   - Computer networks, distributed systems
   - Human-computer interaction, involving communication between computers and people
   - Operating systems: the control of computers, resources and interactions
4. Practice:
   - Problem identification and analysis
   - Design, development, testing and evaluation
   - Management and organization
   - Professionalism and ethics
   - Commercial and Industrial exploration
5. Theory:
   - Algorithm design and analysis
   - Formal methods and description techniques
   - Modelling and frameworks
   - Analysis, prediction and generalisation
   - Human behaviour and performance

1.1.4 Computing Careers

Computing professionals might find themselves in a variety of environments in academia, research, industry, government, private and business organizations – analyzing problems for solutions, formulating and testing, using advanced communications or multi-media equipment, or working in teams for product development. Here's a short list of research and vocational areas in computing (IEEE Computer Society [www.computer.org], www.computer.org/education/careers.htm).

1. Artificial Intelligence – Develop computers that simulate human learning and reasoning ability.
2. Computer Design and Engineering – Design new computer circuits, microchips, and other electronic components.
3. Computer Architecture – Design new computer instruction sets, and combine electronic or optical components to provide powerful but cost-effective computing.
4. Software Engineering – Develop methods for the production of software systems on time, within budget, and with few or no defects.
5. Computer Theory – Investigate the fundamental theories of how computers solve problems, and apply the results to other areas of computer science.
6. Operating Systems and Networks – Develop the basic software computers use to supervise themselves or to communicate with other computers.

7. Information Technology – Develop and manage information systems that support a business or organization.

8. Software Applications – Apply computing and technology to solving problems outside the computer field - in education or medicine, for example.

In summary areas within the fields such as Artificial Intelligence, Computer Science, Information Systems, Hardware Engineering, Software Engineering, Multi-media and Networks form familiar domains of activity which are represented strongly within Computing (Computing; ISBN 185824 489 7; ©Quality Assurance Agency for Higher Education 2000; Published by Quality Assurance Agency for Higher Education, Southgate House, Southgate street, Gloucester GL1 1UB, Tel 01452 557000, Fax 01452 557070, web www.qaa.ac.uk; Printed by Kall Kwik, Gloucester.)

2 The Program

Software Engineering is the systematic development and application of techniques which lead to the creation of correct and reliable computer software. To achieve that, you need a wide understanding of the general principles which underpin not only computer software but also computer hardware and computer communications. As our daily lives become more dependent on computer systems, it is vital that such systems are error-free and totally reliable. This reliability is particularly important when computers are used in safety-critical situations such as hospitals, or controlling aircraft or industrial plants. Software Engineering focuses on challenging problems in industry and commerce concerned with software development and reliability. With concern for software reliability, correctness, safety and cost, such concern being the hallmark of Software Engineering. The principles and knowledge of computer science, engineering, and mathematical analysis are employed by computer software engineers for designing, developing, testing, and evaluating the software and the systems that computers use to carry out various applications.

Software engineers are engaged in analyzing user needs and designing, constructing, testing, and maintaining computer applications software or systems. Various kinds of software like software for operating systems and network distribution, and compilers, which convert programs for execution on a computer, are developed by a software engineer. In the programming or coding fields, software engineers give instructions to a computer, line by line; on how on perform a function or operation. These engineers are also geared to tackle technical problems and hitches. Although these engineers need to possess string programming skills, they are more occupied with the development of algorithms and in analyzing and solving problems in programming than with writing codes. Computer software engineers are usually a part of the team that designs and develops advanced hardware, software, and systems. Thus, until a finished product is developed and released, workers from various branches including those of engineering, marketing, production and design collaborate with each other, of which software engineers are a basic part.

1. Computer applications software engineers – are engaged in analyzing user needs as well as designing, constructing, and maintaining computer applications software and specialized utility programs. Various programming languages are used by these engineers, which are chosen regarding the required purpose for which a computer program would be used. C and C++ are the programming languages that are most commonly while Java, with Fortran and COBOL are used less extensively. Either packaged systems and software systems or specific customized applications are designed and developed by software engineers.

2. Computer systems software engineers – are involved in coordinating the construction of the computer systems of an organization, maintaining them and planning their future growth. They work with a particular company and coordinate the needs and demands of the computer needs of every department by ordering, inventory, billing, and payroll recordkeeping. Suggestions are also made about a computer systems' technical direction. A company's intranet (the network which links computers inside a organization and ease communication among the various departments) is also constructed by these engineers. Systems software engineers work for companies that need configuration, implementation, and installation of complete computer systems. These engineers may also be part of the marketing or sales staff, and serve as the

chief technical resource for these sales officers, staff, as well as customers. They may even engage in product sales and provide continued technical support to the buyers and consumers.

2.1 The B.Sc. in Software Engineering Degree Programme

This programme offers a course of study leading to the B.Sc. in Software Engineering.

2.1.1 Objectives

The objectives of the B.Sc. in Software Engineering programme are: -

1. To produce graduates who are well-educated in the fundamental concepts of software engineering and able to continue their professional development throughout their careers. The course combines theory with consideration of its application in software engineering practice.
2. To build human resource capacity in the Software engineering discipline in both the public and private sectors to students who wish to become proficient in developing software in a variety of languages, platforms and applications using a methodical approach.
3. To produce graduates with good communication skills capable of functioning responsibly in diverse environments and able to work in teams.
4. To produce graduates who are innovative and are capable of creating jobs;

2.1.2 Justification

The Bachelor of Science in Software Engineering Degree Programme will be offered to give an opportunity to prospective students to undertake training in Software engineering at a bachelor's degree level within Uganda. With the growth of the ICT sector in Uganda, there is need to produce graduates who are job creators using the skills obtained in software Engineering .

2.2 Admission Requirements

Admission to the B.Sc. in Software Engineering (B.Sc.SE) degree course will be through three avenues; Direct entry, Mature age and Diploma entry schemes.

To be admitted for a course leading to the award of Bachelor of Science in Software Engineering (B.Sc.SE.) Degree, a candidate must satisfy the general minimum entrance requirements of Makerere University. In addition, the following regulations shall hold for the B.Sc.SE. Degree: -

2.2.1 Direct Entry

Candidates seeking admission through this avenue must have obtained: -

1. At least two principal passes at the same sitting in Uganda Advanced Certificate of Education (UACE) of which one of them must be in either Mathematics or Physics.
2. A minimum weighted points set by the Admissions Board. For purposes of computing weighted points, the advanced level subjects shall be grouped and weighted as follows: -

Group	Weight	Subjects
Essential	3	Two best done of the following subjects: Mathematics, Physics, Chemistry, Economics, Geography, Biology and Fine Art.
Relevant	2	The third best done of the following subjects: Mathematics, Physics, Chemistry, Economics, Geography, Biology and Fine Art.
Desirable	1	General Paper, Sub-Mathematics.
Others	1/2	All others.

2.2.2 Mature Age Entry Scheme

For admission under the Mature Age Entry Scheme, a candidate must have passed the Makerere University Mature Age Entry Examinations.

2.2.3 Diploma Holders

Applicants should possess at least a second class (lower division) Diploma in Computer Science, Engineering, Statistics or any other diploma with either Mathematics or Computer Science, as one of the subjects from any recognised Institution.

2.3 Duration

The degree programme will extend over a period of Four (4) years. An academic year shall consist of two semesters of 17 weeks (15 weeks for classes and 2 weeks for examinations). The first, second and third years will in addition have a recess term of 10 weeks. A full-time student shall not carry less than 15 credit units and not more than 25 credit units per semester.

2.4 Study Format

There will be a day program that will have government sponsored and privately sponsored students. There will also be evenning program that is privately sponsored.

2.5 Distinctive Features

The BSE program has been designed to be hands-on where students acquire practical skills. The program is a four year course and at the end of the first year, each student undertakes a Professional Software Engineering Mini Practical Project. The Mini Project is completed in the recess term of the second semester. The project provides a major opportunity for creativity and serious independent learning.

The Mini Project is introduce early (year 1) such that the student has enough time to practice and improve on the project as the years progress. In year 3, students are exposed to the industry for internship leading to a final software engineering project that is carried out in year 4. The final software engineering project spreads over part of year 4 semesters 1 and two.

2.6 Iternship and Experimental Learning

Year 1 and Year 2 recess terms involves two Professional Mini Projects that are carried out at the university. Each student is assigned a mentor and to look at the technical details of the project. The project may be from the industry.

In Year 3 recess term, the students are attached to the industry for industrial training. Each student has an industrial supervisor and an academic mentor who is a member of staff. Each student is visted at least twice at the place of attachment.

2.7 Teaching and Learning Methods

The mode of delivery will be mainly face-to-face through interactive lectures, group sessions, guided discovery and creativity exercises. The teaching will grounded to problem based activities

2.8 B.Sc.SE Graduate Profile

Software Engineers from the Makerere University, Faculty of Computing and Information Technology will be specialists in information systems for companies, distributed systems design and development, design and development Embedded systems development and web technologies Above all, they will

be professionals with multi-discipline skill which will allow them to fulfil the expectations of the labour market. The graduates of this programme will be equipped with business development and management skills which are vital for the creation of new opportunities and innovations. The graduates will also be equipped with ethical and moral values as engineers, encouraging them to behave as responsible citizens in the practice of their professional, vital for the development of a morally upright society. A typical graduate of the programme will have the following attributes:

1. Ability to use mathematics to model physical components and systems;
2. Ability to design and conduct experiments, as well as analyse and interpret data;
3. Ability to analyse and design software systems using software engineering to meet functional requirements subject to specifications;
4. Ability to utilise software packages for tasks such as: data collection and analysis, computer-aided design, system simulation and analysis, and effective technical communications;
5. Ability to communicate effectively with both technically trained and non-technically trained personnel, through oral presentations and written documents, including appropriate graphics;
6. Ability to work effectively in teams, whether composed of all technically trained or a combination of technically trained and non-technically trained personnel;
7. Ability to structure and analyse situations which pose problems amenable to technical solutions, to define possible alternative approaches to such solutions, to evaluate these alternatives and select the most feasible, subject to institutional, economic, environmental, and other social constraints;
8. Ability and appreciation for developing, both professionally and personally, through lifelong learning;
9. Ability to analyse and design digital and computer communication systems;
10. Ability to analyse and design systems that satisfy system-level objectives by exploiting the synergism of hardware and software through their concurrent design;
11. Ability to specify and design digital hardware interfaces with appropriate software modules for communicating with and control of the interfaced application;
12. Ability to use contemporary engineering design tools and high-level development environments to analyse and design complex systems containing software;
13. Ability to analyse and design systems that satisfy system-level objectives by exploiting operating system services and networking services; and
14. Ability to model the discipline of computing using the topical areas of programming languages, information retrieval, numeric and symbolic computation, algorithms, data structures, digital systems, computer organization, interfacing, architecture, software methodology and engineering, and operating systems.

3 Resources

3.1 Human Resource (Academic Staff)

The Faculty of Computing and Information Technology has sufficient academic staff to run this programme. The Faculty is already running a Bachelor of Computer Science and a Bachelor of Information Technology which programmes share a number of courses with the proposed programme as indicated in the course outline. Through Nuffic, CIT has got a four years 5.7 million Euro grant (June 2007 March 2011) from the Netherlands Programme for the Institutional Strengthening of Post - Secondary Education and Training Capacity (NPT). Under this project there are funds for supporting eight (8) visiting academic staff (under sabbatical) per year for four years. 1.5 million

Euros is for short-term staff missions of 3-6 months for teaching and research activities in Uganda for the whole project period. The staff from Universities in the Netherlands will participate in running graduate programmes in the Faculty.

Makerere University Faculty of Computing and Information Technology uses over 20 PhD holders (senior staff) within the faculty, staff from outside the faculty like in other faculties and institutes within Makerere University, staff from other universities within Uganda, and PhD holders in ICT working with the public and private sectors in Uganda. CIT has over 40 M.Sc. holders on the university payroll at the rank of (Assistant) Lecturer of which 15 are in the Department of Networks. CIT has over 50 PhD students in 2nd and 3rd year of which 18 are registered at Makerere University and they conduct tutorials for graduate students. CIT also recruited over 40 PhD students in first year of 2007/2008 academic year who will also provide tutorials to both undergraduate and postgraduate students. In addition the following have just been appointed as full time senior lecturers in the faculty by the university appointment board:Jude T. Lubega (PhD Computer Science, University of Reading, UK, 2006). Jose Ghislain Quenum (PhD Computer Science, Pierre and Marie Curie University, France, 2005). He joined the Faculty in June 2007 as a Senior Lecturer. Martin Bagaya (PhD Information Systems, Nova Southeastern University, USA, 2006). He joined the Faculty in September 2007 as a Lecturer. John Quinn (PhD Computer Science, University of Edinburgh, Scotland, 2007). He joined the Faculty in September 2007 as a Lecturer. Idris A. Rai (PhD Networks & Computer Science, ENST/Institute Eurecom, France, 2004). He joined the Faculty in October 2007 as an Associate Professor in the Department of Networks. Besides, Prof. Janet Aisbett, a Professor of Information Systems and Computer Science at the University of New Castle; Prof. Greg Gwynn Gibbon, a Professor of Information Systems at the University of Newcastle; and Prof. HN Muyingi, a Professor of Computer Science at Fort Hare University in South Africa joined the faculty as visiting fellows for at least one year. For a detailed list of academic staff see Appendix B on 41.

3.2 Physical Facilities

CIT has two buildings that can accommodate up to 10,000 students in one sitting. The new building (proposed to be called CIT Block B) alone has 14 small and big lecture theatres. Also, for practical-intensive courses, computer labs are used for lecture space.

3.3 Computing Equipment

CIT has setup modern computing laboratories for all its undergraduate and graduate students and computer to student ratio now stands at 1:1. CIT has acquired more than 3000 computers and an assortment of ICT equipment under the Project on Building a Sustainable ICT Training Capacity in the four Public Universities in Uganda.

In the new CIT building there are: 6 large computer labs each accommodating 700 students of which one is fully equipped; 4 smaller computer labs each accommodating 120 students of which two are fully equipped. The new building has a Desktop publishing Unit and specialized labs such as Multimedia Lab, Advanced GIS Lab, Mobile Computing lab, Software Incubation lab, E-Learning Lab, Multi-Media Studio, software engineering lab; and Computer Engineering lab.

3.4 Library

CIT is equipped with a library that offers reading services and textbook loan services. The reading services that are offered within the library premises cater for a maximum of 50 occupants who can make use of the services on weekdays from 8am to 5pm. In order to offer maximum utilization of at least 5,000 volumes of textbooks maintained by the CIT librarian, loan (borrow and return) services are offered to both students and staff. The CIT library acquires textbooks from purchases made by the Faculty and Makerere University Main Library. By the end of 2009, CIT will be equipped with a digital library so as to enhance access to academic content through the use of information and communication technology (ICTs). Makerere University Library services complement CIT library services by offering online library services that include an online catalogue and a variety of electronic resources that support research in computing and information technology. For more information

about the Makerere University Library, checkout the website: http://mulib.mak.ac.ug/

3.5 Financial Resources

Tuition fee per student shall be 2,600,000 Uganda Shillings per annum for Ugandans and 2,500 US Dollars per annum for Non-Ugandans. An extended budget is provided in Appendix A on 39.

3.6 Administration and Technical Support

The department has a head of department assisted by two assistant heads. One assistant head of departments focuses on the academic programs to monitor and manage the teaching process. He/She will oversee on daily basis the actual teaching on the program. The second head of department oversee short certificate courses and community out rich. He/she will help to facility industrial training and placement of students for the projects. Further, the department has administrative assistant to help on different queries that may be raised by students.

To further support community and private partnership the Faculty has a Cooperate Relations Office that links students and staff to the public. In addition, the Faculty has a Workforce development personal to help and link students to potential employs.

In relation to practices and Labs, the faculty has a dedicate ICT services unit to ensure that all ICT facilities are up and running. For student who have their own laptops, there is a wireless access area which is accessible by students anytime. Moreover, each of the Labs has a dedicated Lab assistant and Computer engineer to ensure help students during there free time outside lecture hours. The lab facilities and lab assistants are available 24 hours 7 days a week.

4 Regulations

Here we give regulations specific to the programme. Additional normal regulations that relate to illiness, absence from the program conduct of examinations can be found in the undergraduate handbook available at Academic registrars office and Faculty of Computing and Information Technology.

4.1 Course Assessments

a) Each Course will be assessed on the basis of 100 total marks with proportions as follows:
Course Work - 40; and
Examination - 60.
b) A minimum of two Course Assignments/Tests shall be required per Course.

4.2 Grading of Courses

a) Each Course will be graded out of a maximum of 100 marks and assigned an appropriate letter grade and a grade point as follows:

MARKS %	LETTER GRADE	GRADE POINT
80 - 100	A	5.0
75 - 79.9	B+	4.5
70 - 74.9	B	4.0
65 - 69.9	B-	3.5
60 - 64.9	C+	3.0
55 - 59.9	C	2.5
50 - 54.9	C-	2.0
45 - 49.9	D+	1.5
40 - 44.9	D	1.0
35 - 39.9	D-	0.5
Below 35%	E	0.0

b) The following additional letters will be used, where appropriate: -

W - Withdraw from Course;
I - Incomplete;
AU - Audited Course Only;
P - Pass;
F - Failure.

4.3 Minimum Pass Mark

A minimum pass grade for each course shall be 2.0 grade points.

4.4 Calculation of Cumulative Grade Point Average (CGPA)

The CGPA shall be calculated as follows: -

$$\text{CGPA} = \frac{\sum_{i=1}^{n}(GP_i \times CU_i)}{\sum_{i=1}^{n} CU_i},$$

where GP_i is the Grade Point score of a particular course i; CU_i is the number of Credit Units of course i; and n is the number of courses so far done.

4.5 Progression

Progression through the programme shall be assessed in three ways:

4.5.1 Normal Progress

This occurs when a student passes each course taken with a minimum Grade Point of 2.0.

4.5.2 Probationary

This is a warning stage and occurs if either the cumulative grade point average (CGPA) is less than 2.0 and/ or the student has failed a core course. Probation is waved when these conditions cease to hold.

4.5.3 Discontinuation

When a student accumulates three consecutive probations based on the CGPA or the same core course(s), he/she shall be discontinued.

4.6 Re-taking a Course

A Student may re-take any course when it is offered again in order to pass if the student had failed the course. A Student may take a substitute elective, where the Student does not wish to re-take a failed elective.

4.7 Graduation Requirements

To qualify for the award of the degree of Bachelor of Science in Software Engineering, a candidate is required to obtain a minimum of 167 credit units for courses passed including all the compulsory courses and required number of elective courses within a period stipulated by the University Senate and Council.

5 Curriculum

5.1 Weighting System

The weighting unit is a credit unit. One credit unit is one contact hour per week per semester. One contact hour can be defined as follows: -

- 1 Lecture Hour (LH) is equivalent to 1 Contact Hour (CH).
- 2 Tutorial Hours (TH) are equivalent to 1 contact hour.
- 2 Practical Hours (PH) are equivalent to 1 contact hour.

All the students must make extensive use of the computing facilities outside the scheduled lecture, tutorial and practical hours. The details of the course structure is shown below, where LH, TH, PH, CH and CU stand for Lecture Hours, Tutorial Hours, Practical Hours, Contact Hours and Credit Units respectively.

5.2 Course Structure

CODE	COURSE TITLE	LH	TH	PH	CH	CU	Status
YEAR 1:	**SEMESTER 1:**						
Cores:-	**(6 Core Courses)**						
CSK 1101	Communication Skills	30	30	–	45	3	Current
CSC 1100	Computer Literacy	45	–	30	60	4	Current
BIT 1102	Communications Technology and Internet	45	–	30	60	4	Current
BSE 1103	System Analysis And Design	30	30	–	45	3	Current
BSE 1104	Discrete Mathematics	30	30	–	45	3	New
BSE 1105	Business Law	30	30	–	45	3	New
Electives:-	**(No Elective Course)**						
YEAR 1:	**SEMESTER 2:**						
Cores:-	**(6 Core Courses)**						
BSE 1200	Software Development Principles I	45	–	30	60	4	New
BSE 1202	Principles of Programming I	45	–	30	60	4	Current
BSE 1201	Numerical Analysis	30	30	–	45	3	New
CSC 1203	Information Systems	30	30	–	45	3	Current
BSE 1205	Introduction to Internet Programming	45	–	30	60	4	New
CSC 1204	Research Methodology	30	30	–	45	3	Current
Electives:-	**(No Elective Course)**						
YEAR 1:	**RECESS TERM:**						
BSE 1301	Proffesional Software Engineering Mini Practical Project I	–	–	120	60	4	New
YEAR 2:	**SEMESTER 1:**						
Cores:-	**(6 Core Courses)**						
BSE 2101	Introduction to Database Systems	45	–	30	60	4	New
BSE 2102	Software Development Principles II	45	–	30	60	4	New
BSE 2103	Principles of Programming II	45	–	30	60	4	Current
CSC 2100	Data Structures and Algorithms	45	–	30	45	4	Current
BSE 2104	Computer Architecture	30	30	–	45	3	Current
BSE 2105	Formal Methods	30	30	–	45	3	New
Electives:-	**(No Elective Course)**						
YEAR 2:	**SEMESTER 2:**						
Cores:-	**(5 Core Courses)**						
BSE 2200	System Software	45	–	30	60	4	new
BSE 2201	Network Application Development	45	–	30	60	4	New
BSE 2202	Embedded Systems Software	45	–	30	60	4	New
BSE 2203	Computer Networks and Data communication	45	–	30	60	4	New
BSE 2204	Modelling in Information Systems	45	–	30	60	4	New
Electives:-	**(No Elective Course)**						
YEAR 2:	**RECESS TERM:**						
BSE 2301	Proffesional Software Engineering Mini Practical Project II	–	–	120	60	4	New

CODE	COURSE TITLE	LH	TH	PH	CH	CU	Status
YEAR 3:	**SEMESTER 1:**						
Cores:-	**(5 Core Courses)**						
BSE 3101	IT Projects Management and Development	30	30	–	45	3	New
BSE 3102	Object-Oriented Software Engineering	45	–	30	60	4	New
BSE 3103	Requirements Engineering	45	–	30	60	4	New
CSC 3100	Database Management Systems	45	–	30	60	4	Current
BSE 3104	Software Metrics	30	–	30	45	3	New
Electives:-	**(At least 1 Elective Course)**						
BSE 3105	Software Evolution	45	–	30	60	4	New
BSE 3106	Mobile Networks and Computing	45	30	–	60	4	Current
BSE 3107	Real-Time and Embedded Systems	45	–	30	60	4	New
YEAR 3:	**SEMESTER 2:**						
Cores:-	**(4 Core Courses)**						
BSE 3201	Software Architecture	45	–	30	60	4	New
BSE 3202	Distributed Systems Development	45	–	30	60	4	Current
BSE 3203	Object-Oriented Programming	45	–	30	60	4	Current
BSE 3204	Human-Computer Interface Design	45	–	30	60	4	Current
Electives:-	**(At least 1 Elective Course)**						
BSE 3205	Unix Shell Programming	45	–	30	60	4	New
CSC 3202	Concepts of CAD/CAM	45	–	30	60	4	Current
BSE 3206	Computer Graphics	45	–	30	60	4	Current
YEAR 3:	**RECESS TERM:**						
BSE 3301	Internship	–	–	120	60	4	Current
YEAR 4:	**SEMESTER 1:**						
Cores:-	**(All Core Courses)**						
BSE 4100	Software Engineering Project I	–	–	150	75	5	New
BSE 4101	Software Reliability and Testing	45	–	30	60	4	New
BSE 4102	Ethics for Proffesional Engineers	30	30	–	45	3	New
BSE 4103	Entrepreneurship and Business	30	30	–	45	3	New
YEAR 4:	**SEMESTER 2:**						
Cores:-	**(All Core Courses)**						
BSE 4200	Software Engineering Project II	–	–	150	75	5	New
BSE 4201	Software Design Patterns	45	30	–	60	4	New
BSE 4203	Information Technology and Society	30	30	–	45	3	New
BSE 4204	Emmerging trends in software engineering	30	30	–	45	3	New

6 Course Unit Descriptions

6.1 Year 1 Semester I

6.1.1 CSK 1101: Communication Skills (3 CU)

Course Objectives Upon successful completion of this course, the student is expected to have gained the following: (i)Improved communication competencies of student; (ii) Improved problem solving strategies; (iii) Attained the art of critical thinking; (iv) Ability to collect and synthesis information; and (v) Knowledge about utilising the library and other educational resources.

Course Content Writing Skills Thinking critically/ selectively before the writing process; Selecting the relevant details; Organising the relevant details logically; Writing the reports essays, letters and taking notes in appropriate register; Avoiding ambiguities, fallacies, irrationalities; Providing supportive evidence; Editing documents, proof reading; Writing and expanding information; Quoting and citing references; Writing a curriculum vitae. Reading Skills The use of skimming; scanning inference and prediction in reading; Intensive and critical reading; Acquisition of specific reading skills; Interpretation of non linear texts; Locating information and comprehension. Speaking and Listening Skills to Enhance E.ffective Public Relations The art of persuasion in e.effective speaking; Conducting interviews; Conducting meetings; Participating in group discussions and tutorials; Non verbal communication crues; Presentation seminars, seeking clarification etc.; Expression of politeness; Public speaking; Proper listening skills. Examination Skills Preparing for examinations; How much one gets from group discussions; Proper revision; Understanding examination rubric; Budgeting time during examination process; Writing examinations and following instructions.

References

- 101 ways to improve your communication skills instantly, by Bennie Bough, 4th Edition, 2005
- The hard Truth About soft skills: Work Place Lessons Smart People wish they had learned sooner, by Peggy Klavs, 2008

6.1.2 CSC 1100: Computer Literacy (4 CU)

Course Objectives The course aims at preparing students for various fields of Computing by providing them the basic understanding of core concepts. Students are introduced to the field of computing and fundamentals of programming, computers, and program development tools. In addition students will gain ability to use a personal computer for personal and professional purposes as well as use internet technologies like the wide world web and e-mail.

Course Content Introduction to Computers and Computer Systems: Basic computer processing, software categories, digital computers, binary numbers; Computer architecture, input/ output devices, main and secondary memory, central processing unit. Introduction to operating systems (Windows e.g. Windows 98 and Windows NT 4.0, Linux, Unix). Word Processing: Producing documents with formats and styles; Use mail merge to create letters; Manipulate text and graphics and documents; Create templates and forms for use; Import/ Export other documents into MS Word. Spreadsheets and Modelling: Use MS Excel for budgeting and analysing financial data; Use MS Excel for analysing statistical data; Add charts and objects to worksheets; Import data from other applications such as databases; Carry out simulations and modelling using spreadsheets. Lotus. MS PowerPoint: Produce excellent presentations with MS PowerPoint. Others: Core Draw; Photo Paint (Photo draw); Front page; Publisher; Outlook. Web Technology and Networks: HTTP, Hyper Text Markup Language (HTML), the basics of the Internet, Local-Area and Wide-Area Networks, the World-Wide Web (WWW), and Uniform Recourse Locators (URLs). Database management using Microsoft Access.

References

- Computer Literacy, by John Preston, Robert Ferrett and Shelley Gaskin, 2007
- Practical Computer literacy by Jelne Janrich and dan Oja, 2004

6.1.3 BIT 1102: Communications Technology and Internet (4 CU)

Course Objectives Upon successful completion of the course, the student is expected to have acquired the following knowledge elements and skills; (i)An understanding of the operation of networked systems including Local Area Networks, Intranet, Internet Communication Protocols, and e-services; (ii)Internet infrastructure and governance; (iii)Social, Moral, legal and ethical aspects of digital communication; (iv)Basic internet security assurance principles; (v)In-depth understanding of the operation of web browsers; and (vi)Opportunities and threat to business and development due to advancement of digital communication technologies.

Course Content Topics include networking concepts, Internet and intranet tools, protocols and security. Also included are the infrastructure and governance of the Internet, with emphasis on personal, business, social, legal and ethical implications. Recommended skills are keyboarding and experience with e-mail and web browser software.

References

- Network Security Essentials, Applications and standards, by William Stallings, Prentice Hall, 2nd, 2005
- Using the Internet: User Friendly Reference, By William Eager and Bill Eager, Que publisher, 1994, Original from the University of California, ISBN 0789700964, 9780789700964
- Online Communication: Linking Technology, Identity, and Culture; By Andrew F. Wood and Matthew J. Smith, Routledge publisher, 2005, ISBN 0805848495, 9780805848496
- Communication Technology and Social Change: Theory and Implications; By Carolyn A. Lin and David J. Atkin, Routledge publisher, 2007, ISBN 0805856145, 9780805856149

6.1.4 BSE 1103: System Analysis and Design (3 CU)

Course Objectives Upon successful completion, the student should be able: (i)Apply appropriate techniques to different stages of software development life cycle; (ii)Have hands-on experience with software development tools for systems analysis and design; and (iii) Use analytical skills to structure complex systems into manageable pieces.

Course Content The System Development Process (Steps of Systems Analysis); Project Selection (Sources for Ideas for Projects, Selection Criteria); Feasibility Study: Cost-Benefit Analysis, Definition Phase, Analysis Phase, Design Phase, Implementation Phase, Evaluation Phase; Review. Techniques: Fact Gathering- Interviewing and Others; Charting Techniques: Procedure Charts, Flowcharts, HIPO Chart, Others; Decision Tables; Simulation: Desk Simulation, Benchmarks, Simulation Entirely by Software. General Systems Considerations: Data Capture and Output; Data Management and Security; Methods and Procedures; Data Communications; Systems Maintenance; User Involvement. Project Management: Problems of Implementation- - Personal Training, File Creation and Conversion, Programming, System Testing, Design Changes, Planning for Implementation; Management Control- Control over Systems Development, Project Control. Case Studies.

References

- Systems Analysis and Design: A Structured Approach by William S. Davis, 1983.
- Systems Analysis and Design Methods by Jeffrey L. Whitten, Lonnie D. Bentley, and Victor M. Barlow, 1989.

6.1.5 BSE 1104: Discrete Mathematics (3 CU)

Course Objectives Upon successful completion of this course, the student will: (i) Be familiar with the terminology, operations, and symbols of set theory, and with formal logic. (ii) Be able to use logic to determine the validity of an argument. (iii) Be able to construct the proof of a theorem directly, by the contrapositive, by cases, by contradiction, by truth table, by counter-example, and by mathematical induction. (iv) Be able to identify a relation; specifically, a partial order, equivalence relation, or total order. (v) Be able to identify a function; specifically, surjective, injective, and bijective functions. (vi) Be able to perform operations on matrices. (vii) Be familiar with the terminology for graphs and trees. (viii) Be able to trace Euler and Hamiltonian paths. (ix) Be able to construct minimal spanning trees and adjacency matrices for graphs. (x) Have begun to develop a logical mode of thought that will be applicable to computer design, both hardware and software. (xi) Be able to understand sequential logic (xii) Be able to understand sets and relations (xiii) Be able to represent information using zeros and ones.

Course Content This course provides an introduction to several topics fundamental to computer science. Topics discussed include set algebra, logic, relations and functions, recursion, matrices, graph theory, and methods of proof. Emphasis is on an algorithmic approach. Set theory; Methods of proof; Recursion; Matrix algebra; Graphs and trees. Application to data structure and graph representations, partial ordered sets, trees, algebraic structures, lattices and Boolean algebra, semi groups, groups, introduction to grammars and machines and languages, error correcting codes.

Representation of information, two's complement arithmetic. Combinational logic: switching algebra, canonical forms, Karnaugh maps, combinational network analysis and design, MSI modules. Sequential logic: latch, flip-flop and logic design, state diagram, sequential network analysis and synthesis, register, counter, memory organization.

References

- Bobrow, L.S. and Arbib, M.A. Discrete Mathematics: Applied Algebra for Computer and Information Science. Philadelphia, PA: Saunders, 1974.
- Dossey, J.A.; Otto, AD.; Spence, L.; and Eynden, C.V. Discrete Mathematics, 3rd ed. Reading, MA: Addison-Wesley, 1997.
- Balakrishnan, V.K. Introductory Discrete Mathematics. New York: Dover, 1997.

6.1.6 BSE 1105: Business Law (3 CU)

Course Objectives Upon successful completion of this course, the student will: (i) Understand the basic nature of the legal system including the court structure and the role of lawyers (ii) Understand the different types of torts and crimes (iii) Be familiar with contract law including all the elements of a contract under the common law as well as the Uniform Commercial Code (iv) Understand the basic theories of products liability law (v) Be familiar with the creation of security interests and the rules that secured creditors must follow (vi) Understand the basic features of bankruptcy law (vii) Understand the general nature of an agency relationship

Course Content This subject acquaints the student with legal concepts and their application to business and personal situations. Attention is paid to problems arising under the following topical headings: basic nature of the legal system; tort law; contract law, including both common law principles and the provisions of the Uniform Commercial Code; products liability law; debtor/creditor relations; bankruptcy law; and agency law.

Introduction to Legal System; Tort and Criminal law; Contract Law; Uniform Commercial Code; Products Liability Law; Secured Transactions; Bankruptcy; Agency.

References

- Uganda Business Law Handbook, by Ibp Usa and Emerging Markets Investment Center, International Business Publications, USA; 2 edition (May 5, 1999), ISBN-10: 0739705768

6.2 Year I Semester 2

6.2.1 BSE 1200: Principles of Software Development I (4 CU)

Course Objectives Upon successful completion, the student should be able to: (i) Demonstrate mastery of the software development process, describing core activities at various levels of software development; (ii) Perform requirement analysis of a simple software systems; (iii) Manage a team of developers; (iv) Demonstrate knowledge of at least one tool of software documentation; (v) Work in a team; and (vi) Design, develop, test and valid a software product.

Course Content Topics covered include: Introduction: What is software engineering? Phases in the development of software. Software Management: Planning and controlling a software development project. Requirements analysis: Requirements specification, humans as information sources, tools for documenting the requirements (SADT, PSL/PSA, Ada-based analysis methods). Software design:Abstraction, modularity, information hiding, design methods, design documentation. Testing: Test objectives, verification and validation, manual test techniques, techniques for functional and structural testing.

References

- Balancing Agility and Discipline: A Guide for the Perplexed; by Barry Boehm and Richard Turner, Addison Wesley Professional, 2003, ISBN-10: 0321186125.
- 201 Principles of Software Development, by Alan M. Davis, McGraw-Hill, March, 1995, ISBN-10: 0070158401

6.2.2 BSE 1201: Numerical Analysis (4 CU)

Course Objectives Upon completion of the course, the student should be able to: (i)Demonstrate factual knowledge including the mathematical notation and terminology used in the course; (ii) Describe the fundamental principles including the laws and theorems arising from the concepts covered in this course; (iii) Apply course material along with techniques and procedures covered in this course to solve practicle problems; and (iv) Write numerical programs, such as Matlab programs, to solve the above problems.

Course Content Numerical linear algebra, numerical solution of systems of non-linear equations, approximations, Fast Fourier-Transformation, numerical integration, di.erence equations and numerical solution of ordinary differential equations. Problem solving is an important part of the course.

References

- Numerical Analysis, by Burden, R.L. and Faires, D.J, Eighth Edition,Brooks/Cole Publishing Co., Pacific Grove, CA, 2001. ISBN: 0-534-38216-9

6.2.3 BSE 1202: Principles of Programming I (4 CU)

Course Objectives Upon completion of the course, the student should be able to: (i)Demonstrate mastery of basic programming concepts, like program structure and control, Application Program Interface (API), memory management among others; (ii)Demonstrate sufficient understanding of Java as a programming language; (iii)Develop simple web and stand alone applications in Java; (iv)Work with an Integrate Development Kit like NETBEANS with minimal difficulties; and (v)Differentiate the various programing languages (i.e., in terms of their strength and weakness).

Course Content This course is an introduction to the Java programming language. The course will include an introduction to the concepts of object oriented programming and will show how Java supports this programming paradigm. Students will learn about the Java environment and will write both applets (programs that execute in a Web browser) and applications (stand alone program). In addition to learning about basic language statements, students will also learn how Java provides support for such diverse applications as Web pages, multimedia, educational, etc

References

- Concepts of Programming Languages, by R.W. Sebesta, 5th Edition, Addison Wesley, 2002.
- Progamming Languages: Concepts and Constructs. By R. Sethi , 2nd Edition. Addison Wesley. 1996.
- Introduction to Java Programming: Comprehensive Version, by Y. Daniel Liang, Prentice Hall, 2006, ISBN 0132221586, 9780132221580

6.2.4 BSE 1204: Introduction to Internet Programming (4 CU)

Course Objectives On successfully completing this course, students will be able to: (i) Set up a .Net development including environment, including MSDE; (ii) Apply core ASP.NET technologies to develop Web applications; (ii) Author server-side ASP.NET code in C#; (iv) Create Web applications using ADO.NET to interact with SQL Server; and (v) Write ASP.NET pages that integrate into traditional DHTML-driven pages.

Course Content This course covers ASP.NET development fundamentals, providing hands-on experience through the building of a fully functional time and expense tracking application like TimeEx. Starting with an overview of server-side technologies, it then goes into the C# (C Sharp) programming language commonly used for ASP.NET and database development utilizing MSDE (Microsoft SQL Server Desktop Engine). Particular attention is given to techniques for integrating ASP.NET pages with the advanced client-side (DHTML) technologies covered first

References

- Build Your Own ASP.NET Website Using C# &VB.NET, by Zak Ruvalcaba, 1st Edition, ISBN: 0-9579218-6-1, 2005
- Internet & World Wide Web: How to Program by Paul J. Deite and Harvey M. Deitel, ISBN-10: 0131752421, Prentice Hall; 4 edition , September, 2007
- ASP.NET Unleashed: unleashed By Stephen Walther, Sams Publishing, 2003 ISBN 067232542X, 9780672325427
- Internet Programming with VBScript and JavaScript (Web warrior series), by Kate Kalata, Course Technology, 1st edition, December, 2000. ISBN-10: 0619015233

6.2.5 CSC 1203: Information Systems (3 CU)

Course Objectives On completion of this course, the students should have achieved the following: (i) An understanding of characteristics of information systems and their management; (ii) Technology and its effective utilization within organizations; (iii) Skills in evaluating and applying appropriate methods and technologies for representing, managing and disseminating information; and (iv) Skills in contributing to information system implementation using sound principles, an appreciation of the necessary legal, ethical and professional values; (v) An understanding of information services; (vi) Understanding of principles of secure computing; (vii) Core considerations of information systems acquisition and adaption; and (viii) Basic project management skills.

Course Content Information Systems fundamentals; The role of Information Systems in an organisation; The shift from Data Processing Systems via Information Systems to Knowledge-Based Systems; Information Systems boundaries; How information and knowledge add value to an organization; The Information Technology component of Information Systems; Information Systems Engineering. Organizational strategies and objectives; Strategic planning; Human resource development; Budgeting, costing, charging, hire versus buy decisions; Hardware and software procurement: request for proposal, evaluation, selection, contracting and maintenance; Security in computing: continuity of processing, controls and planning for standby; Computer audit; Project management: approaches, tools, site planning and installation; Implementation strategy; The Information Center; Case studies.

References

- Fundamentals of Information Systems, by Ralph Stair and George Reynolds, Course Technology; 5th edition, Jan, 2009. ISBN-10: 1423925815
- Management Information System, by Kumar N, Anmol Publications PVT. LTD.ISBN 8126116757, 9788126116751
- Information Systems Today: Managing in the Digital World, by Leonard Jessup and Joseph Valacich, Prentice Hall, 3rd edition, April, 2007. ISBN-10: 0132335069
- Introduction to Information Systems: Essentials for the Internetworked Enterprise By James A. O'Brien, Irwin/McGraw-Hill, 2000, ISBN 0072297492, 9780072297492

6.2.6 BSE 1205: Research Methodology (3 CU)

Course Objectives Upon successful completion of the course, the student should be able to: (i)Identify interesting problems of research and develop appropriate strategies to address the problems identified; (ii)Demonstrate ethical consideration in conduction research; (iii)Show knowledge and understanding of Statistical and non-Statistical Research Methods; (iv)Use statistical packages to analyze data and make scientifically sound conclusions based on the analysis; (v)Interpret, explain, and experiment through writing, oral presentation, various multi-media modes, and in a classroom setting; and (vi)Demonstrate reflective, reflexive, creative, interpretative, evaluative, critical, and analytical thinking skills.

Course Content This course is an introduction to research, statistical analysis, and decision making. Close attention is paid to data types, data contributions, the identification of variables, and descriptive data presentation techniques. Students are introduced to both parametric and nonparametric data analysis procedures including independent and dependent sample t-tests, chi-square analysis, and simple analysis of variance. Hypothesis testing and the use of statistical software packages are emphasized.

References

- Essentials of Research Methods, by JANET M RUANE, Montclair State University, Jul 2004, ISBN10: 0631230483

6.3 Year 1 Recess Term

6.3.1 BSE 1301: Proffesional Software Engineering Mini Practical Project I (4 CU)

Course Objectives Upon successful completion of the course, the student should: (i) Demonstrate mastery of a specific programing language and development platform; (ii) Carry out requirement engineering for a particular project; and (iii) Demonstrate ability to develop a mini application or systems with minimal difficulty.

Course Content An introduction to a specific proffesional software certification course is covered.

References

- Accelerated C++: Practical Programming by Example, Andrew Koenig and Barbara E. Moo, Addison-Wesley, January 2008, ISBN 020170353X
- Beginning Java SE 6 Platform: From Novice to Professional, Jeff Friesen, Apress October 2007, 159059830X
- Easy Oracle Jumpstart: Oracle Database Management Concepts and Administration, Steve Karam, and Robert Freeman, ISBN 0-9759135-5-7
- Oracle PL/SQL Language Pocket Reference, by Steven Feuerstein, Bill Pribyl and Chip Dawes, Fourth Edition, October 2007, OReilly, ISBN 10: 0-596-51404-2

6.4 Year 2 Semester 1

6.4.1 BSE 2102: Software Development Principles II (4 CU)

Course Objectives Upon successful completion of the course, the candidate should be: (i)Able to describe the history of software development and software life cycle management; (ii)Able to address various challenges of user interface design and adaption; (iii)Demonstrate ability to carry out system feasibility study and document requirements; (iv)Able to carry out software modularization using one of the most popular software design tools; (v)Show ability to manage a software development process; (vi)Able to design, implement and deploy a software product; and (vii)Able to performance software correctness and testing with minimum difficulties.

Course Content History of software design and development, Software crisis, Software life cycle Management issues, User interfacing, Documentation, Requirements Analysis and Definition, Feasibility study, System specification, Functional software specification Methods, Tools, Design Methodologies: structured design, functional decomposition, data-flow (data-driven), data structure (Jackson), object-oriented design, Tools: abstraction, structure chart, data-flow diagram, program design language, structured walkthrough, Implementation and Testing, Programming environments, teams, languages, and style, Strategies for maintainability and reusability, Test case design, program testing, and system testing, Quality assurance, verification, validation, reliability, Testing methodologies, Evolution,Operation; performance analysis and measurement Maintenance

References

- Balancing Agility and Discipline: A Guide for the Perplexed; by Barry Boehm and Richard Turner, Addison Wesley Professional, 2003, ISBN-10: 0321186125.
- 201 Principles of Software Development, by Alan M. Davis, McGraw-Hill, March, 1995, ISBN-10: 0070158401

6.4.2 BSE 2105: Formal Methods (3 CU)

Course Objectives (i)Demonstrate factual knowledge including the mathematical notation and terminology used in the course; (ii) Be able describe the fundamental principles including the laws and theorems arising from the concepts covered in this course; (iii) Be able to apply course material along with techniques and procedures covered in this course to solve practicle problems; and (iv) Demonstrate programming skills by writting numerical programs, such as Matlab programs, to solve the above problems.

Course Content Predicate Logic Specification: 1. Foundations 2. Basic concepts 3. Verification 4. Z 5. Tools and systems: Z animation – Miranda and ZANS , Nitpick ,the Z Notation. Algebraic Specification: 1. Foundations 2. Basic concepts 3. Verification 4. Tools and systems; Miranda, The OBJ family of languages, LARCH. Optional Topics (as time permits): 1. Statecharts 2. Integrated creation of a program and its correctness proof 3. Automatic program synthesis, Scripting Languages.

References

- Z: An Introduction to Formal Methods, by Antoni Diller, 2nd edition, Wiley, (June 1994), ISBN-10: 0471939730
- Logic in Computer Science: Modelling and Reasoning about Systems, by Michael Huth and Mark Ryan, Cambridge University Press; 2nd Edition (August, 2004), ISBN-10: 052154310X
- Formal Methods and Models for System Design: A System Level Perspective, by Gupta, R., Le Guernic, P. Shukla, S.K. and Talpin, J.P. (Eds.), 2004, ISBN: 978-1-4020-8051-7

6.4.3 BSE 2103: Principles of Programming II (4 CU)

Course Objectives Upon successful completion the students should be able to: (i)Demonstrate mastery of core programming concepts, like program structure and control, Application Program Interface (API), memory management among others; (ii)Demonstrate sufficient understanding of object oriented programming; (iii)Develop complex web and stand alone applications; (iv)Work with an Integrate Development Kit like NETBEANS with minimal difficulties; and (v)Differentiate the various programing languages (i.e., in terms of their strength and weakness)

Course Content Formal definition of structures: formal description of syntax and semantics; Meta - Language; Comparative studies of programming languages and language design concepts: structural organization, structures for names, data abstractions, concurrency; Functional, object-oriented, and logic programming languages and concepts; Language design principles.

References

- Concepts of Programming Languages, by R.W. Sebesta, 5th Edition, Addison Wesley, 2002.
- Progamming Languages: Concepts and Constructs. By R. Sethi , 2nd Edition. Addison Wesley. 1996.
- Introduction to Java Programming: Comprehensive Version, by Y. Daniel Liang, Prentice Hall, 2006, ISBN 0132221586, 9780132221580

6.4.4 CSC 2100: Data Structures and Algorithms (4 CU)

Course Objectives Upon successful completion the students should be able to: (i) Demonstrate factual knowledge including the mathematical notation, syntax and terminology used in the course; (ii) Describe the fundamental principles including the laws and theorems arising from the concepts covered in this course; (iii) Apply course material along with techniques and procedures covered in this course to solve practicle problems; (iv)Design simple algorithms and data structures in C; and (v)Use theories and concepts covered in the course to approximate the complexities of algorithms.

Course Content Methods for modularising, documenting and constructing programs (abstract data types/ object orientation), and an introduction to classical data structures and algorithms with complexity analysis. Detailed Course Content: Application and implementation of strings, arrays, stacks queues, lists, trees and graphs. Abstract data types; storage management. Sorting, searching, merging, reference and cross-referencing files. Design and analysis of algorithms, complexity of algorithms, big-O and small-O notation. Binary trees, B-trees, B*-trees, and AVL-trees. Study of

disk and tape file organizations. Creation of sequential, direct, and indexed sequential files. Hashing techniques and address overflows. Primary and secondary key usage, inverted and linked-list files. Semantic Nets, frames

References

- Data Structures and Algorithm Analysis in C, by Mark weiss, 2nd Edition, Addison-Wesley, (1997), ISBN: 0-201-49840-5
- Data Structures and Algorithm Analysis in Java, by Mark Allen Weiss, Pearson International, ISBN 0-321-37319-7

6.4.5 BSE 2101: Introduction to Database Systems(4 CU)

Course objectives: By the end of the course, students should be able to: (i) Develop a sound data model for an application domain; (ii) Realise this model as a relational database schema; (iii) Implement this schema as a PostgreSQL application; (iv) Build an HTML/forms based (PHP) interface to a relational database; (v) Aanalyse the costs/benefits of data structures and algorithms used in the implementation of relational database management systems; (vi) Understand the theory and techniques behind managing concurrent access to databases and recovery from errors; and (v) Appreciate the limitations of existing relational database technology and have an overview of upcoming database technologies.

Course content: An introduction to SQL, Advanced SQL, Query-By-Example (QBE), Visual Basic and other fourth generation languages will be given. This subject aims to explore the theory behind relational database systems, the practice of developing database applications, and the technologies used to implement database management systems. This course will also cover introduction to Microsoft Access.

References

- Database Systems: The Complete Book, by Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom. Prentice Hall. 2002.
- Fundamentals of Database Systems, by Ramez Elmasri and Shamkant B. Navathe. 4th Edition, ISBN 0-321-12226-7

6.4.6 BSE 2104: Computer Architecture (3 CU)

Course Objectives Upon successful completion of the course, the student should: (i) Have gained an understanding of basic components of the modern computer system; (ii)Be able to describe the operation of the various logic gets; (iii)Be able to design digital circuits; (iv)Demonstrate a good understanding of sequential and parallel processing; (v)Perform low level assembly programming; and (vi) Perform low level memory management.

Course Content Computer Organization and Structures (based on the Von Neumann architecture. Processor unit organization: control unit, ALU, processor register and internal buses; hard-wired and micro-programmed control. Instructions sets, formats and types. Addressing modes, stacks, pipelining, RISC/CISC concepts. Memory organization and addressing; Memory hierarchy and cache. Special-purpose co-processors. I/O facilities and storage devices. The Operating System level.

References

- Computer Architectures: A Quantitative Approach, by D. A. Patterson and J. L. Hennessy, Morgan Kaufmann Publishers, 3rd Edition, 2003.

- Structured Computer Organisation, by Andrew S. Tanenbaum, 4th Edition, January 1999, Prentice Hall inc., Upper Saddle River, NJ 07548 USA, ISBN 0-13-020435-8

6.5 Year 2 Semester 2

6.5.1 BSE 2200: Systems Software (4 CU)

Course Objectives: By the end of this course (i) Students will understand the various levels of system and application software; (ii) They will be familiar with the major Operating System services such as file systems, memory management, process management, device control and network services; (iii) They will understand how design decisions in Operating Systems affect users of the system; (iv) In addition, students will have used a major Operating System extensively, with experience in using an interactive command line programming language; and (v) They will also will have experience in using a systems programming language with an Application Programmers Interface to the Operating System for its services based on Unix OS, and the C systems programming language.

Course content: This unit looks at the necessary system architecture introduction for further study of operating systems, computer architectures, and the implementation of higher level languages. It goes further and builds upon that by looking at the concepts underlying Operating Systems, and to show how different choices in Operating System design and implementation have effects on applications, application programmers and user environments.

References

- System Software: An Introduction to Systems Programming by, Leland L. Beck, Addison Wesley; 3rd edition, August, 1996, ISBN-10: 0201423006
- Modern Operating Systems, by Andrew S. Tanenbaum, 2nd Edition , Prentice Hall 2001, ISBN 0130926418
- Operating Systems Concepts, by Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, 6th Edition, John Wiley & Sons 2002, ISBN 0471250600

6.5.2 BSE 2201 Network Application Development (4 CU)

Course Objectives This course is designed to: (i) Familiarize students with technologies and protocols that support computer communication networks, including the Internet; and (ii) Elaborate on network based programming methodologies, languages, tools and standards.

Course Content Topics include design principles for network-based applications; design and development of Java Servlets, JSP, Web services and .NET; principles of information security in network-based applications; http and https protocols.

References

- Core Servlets and Java Server Pages, Volume 1: Core Technologies by Marty Hall (2nd Edition)

6.5.3 BSE 2203: Computer Networks & Data Communication (4 CU)

Course Objectives Upon successful completion of this course students should be able to: (i) Master the terminology and concepts of the OSI reference model and the TCP/IP reference model; (ii) Master the concepts of protocols, network interfaces, and design/performance issues in local area networks and wide area networks; (iii) Demonstrate knowledge of wireless networking concepts; (iv) Appreciate contemporary issues in networking technologies; and (v) Demonstrate knowledge of network tools.

Course content: Introduction to Networks: definition, advantages, types, configurations; The OSI/ISO reference model; Transmission media: magnetic media, twisted pair, coaxial, fiber-optics; Data encoding: straight, Manchester, differential Manchester, satellite; Digital versus Analog transmission; Modems, modulation and their standards, codes and pulse code modulation; Integrated Services Digital Networks (ISDN); Network Access Protocols; Passive versus dynamic allocation; LAN standards:802.3 (Ethernet), 802.4 (token bus), 802.5 (token ring); Computer Network security, Active and Passive Attacks; Network layer and Network layer protocols; Transport layer and Transport layer protocols. Furthermore, the course considers problems on each layer of a multilayered communication model, and describes some typical solutions to such problems.

References

- James F. Kurose and Keith W. Ross. Computer Networking - A Top Down Approach Featuring the Internet, 3rd edition, Addison-Wesley, ISBN 0-321-22735-2.
- Computer Networks: A Systems Approach. L. Peterson and B. Davie

6.5.4 BSE 2204: Modelling in Information Systems (4 CU)

Course objectives: Upon successful completion of the course, the student should: (i)Be able to demonstrate sufficient understanding of the various modeling tools and techniques; (ii)Be able to model an information system using the Universal Modelling Language (UML); (iii)Be able to model business processes; (iv)Select a suitable modeling language based on the context; and (v)Be able to evaluate the quality of models using model checking tools and formal methods

Course content: This course deals with methods and techniques used in systems analysis and design. This includes modelling the enterprise, application domains, business processes and activities, and information systems components. Main topics to be covered are: Modelling the enterprise: Object-oriented concepts, object-oriented enterprise modelling. Modelling the business process: process and workflow modelling, IDEF0, and SAP R/3 process models, Modelling the dynamics: Data flow diagram (DFD), structured English, decision table, decision tree, and state-transition diagram. Modelling the structures: Entity-relationship (ER) models. Putting the designs together: Unified Modeling Language (UML) Evaluation of methods: Evaluation criteria.

References

- Conceptual Modeling of Information Systems, by Antoni Oliv, Springer, 1st edition (October 1 2007) ISBN-10: 3540393897
- Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations, by Rudy A. Hirschheim, Heinz K. Klein and Kalle Lyytinen, Cambridge University Press (1995), ISBN 0521373697

6.5.5 BSE 2202: Embedded Systems Software (4 CU)

Course Objectives Upon successful completion of this course, the student will: (i) Understand the role of assembly language programming; (ii) Understand the instruction set of a typical embedded processor (68HC11); (iii) Be able to employ a modular approach to assembly language programming with code reuse; (iv) Be able to use embedded systems development tools; (v) Understand memory addressing and use various addressing modes; (vi) Understand hardware interrupts and be able to use them; and (vii) Be able to integrate assembly language subroutines into a high-level language program

Course Content This course presents assembly language programming as the bridge between hardware and high-level programming languages such as C++. Topics covered include the addressing modes, register file, and instruction set of a microcontroller; subsystems such as timers, handshaking

input and output, and analog to digital conversion; and interrupts. Software control of hardware is stressed. In the laboratory, students design software to demonstrate proficiency in these areas.

Introduction to microcomputer/microcontroller structure from a programmers perspective; 68hc11 addressing modes and memory types; Tool usage (assembler, linker, downloader, simulator); 68hc11 instruction set (5 classes) Assembly language programs structure, including comparisons to high-level languages; Parallel I/O with handshaking; Analog I/O on the 68hc11 plus background on A/D and D/A converters; Hardware interrupts.

References

- S. A. Edwards, Languages for Digital Embedded Systems, Kluwer Academic Press, ISBN 079237925X, 2000.

6.6 Year 2 Recess Term

6.6.1 BSE 2302:Proffesional Software Engineering Mini Practical Project II (4 CU)

Course Objectives Same as BSE 1301

Course Content Specific proffesional software certification course covered in BSE 1301 is covered further after which students are required to write an online examination for certification.

References Same as BSE 1301

6.7 Year 3 Semester 1

6.7.1 BSE 3101: IT Project Management and Development(4 CU)

Course objective: At the end of the course, students will: (i) Appreciate the range of practical project management techniques available; (ii) Be able to apply project management principles and techniques within their organisations; and (iii) Be able to confidently undertake the management of a project and see it to a satisfactory conclusion on time and within budget.

Course Content The specific areas to be covered are:

Integration Management includes the processes required to ensure that the various elements of the project are properly co-ordinated. Scope Management is the function of controlling a project in terms of its aims, goals and objectives through the process of conceptual development, full scope definition or statement, scope reporting and control, and project close out. Time Management is the function of maintaining appropriate allocation of time to every element in the overall conduct of the project through the successive stages of its natural life-cycle. Cost Management is the function of maintaining effective financial control of the project. Quality Management is the composite of material attributes (including performance features and characteristics) of the product or service which are required to satisfy the needs for which the project is being undertaken. Human Resource Management is the function of directing and coordinating human resources throughout the life of the project. It involves the application of the principles of behavioural science and administrative knowledge to achieve the predetermined project objectives of scope, time, cost, quality and participant satisfaction. Communications Management includes the processes required to ensure timely and appropriate generation, collection, dissemination, storage and ultimate disposition of project information. Risk Management includes the processes concerned with identifying, analysing and responding to project risk. It includes maximising the results of positive events and minimising the consequences of adverse events. Procurement Management is the function of acquiring resources for the project in order to produce the end product.

References

- Advanced Project Management Techniques, by Dick Billows, PMP

6.7.2 BSE 3104: Software Metrics (3 CU)

Upon successful completion of this course students should be able to: (i) Describe software metrics; (ii) Understand the foundations of measurement theory and models of software engineering measurement; and (iii) Appreciate software products metrics, software process metrics and measuring management.

Course Objectives: The course is composed of the following basic modules: Measurement theory (overview of software metrics, basics of measurement theory, goal-based framework for software measurement, empirical investigation in software engineering), Software product and process measurements (measuring internal product attributes: size and structure, measuring external product attributes: quality, measuring cost and effort, measuring software reliability, software test metrics, object-oriented metrics Measurement management

References

- Software Metrics: A Rigorous and Practical Approach, (2nd ed.) (638p.), N.E. Fenton and S.L. Pfleeger, PWS Publishing, 1998. ISBN 0-534-95425-1. Additional Recommended Text and Reference Books:
- Metrics and Models in Software Quality Engineering, Stephen H. Kan, 2nd ed. (560 p.), Addison-Wesley Professional (2002). ISBN: 0201729156.
- Software Engineering Measurement, John C. Munson, Auerbach Publications, 2003 (443 pages) ISBN:0849315034
- Software Metrics: Measurement for Software Process Improvement, BA Kitchenham, Blackwell Pub, 1996.ISBN: 1855548208.
- Applied Software Measurement: Assuring Productivity and Quality, C. Jones, McGraw-Hill, 1996

6.7.3 CSC 3100: Database Management Systems (4 CU)

Course Objectives Upon successful completion of this course students should: (i) Master the basic concepts and appreciate the applications of database systems; (ii) Master the basics of SQL and construct queries using SQL; (iii) Be familiar with a commercial relational database system (Oracle) by writing SQL using the system; (iv) Be familiar with the relational database theory, and be able to write relational algebra expressions for queries; (v) Mater sound design principles for logical design of databases, including the ER method and normalization approach; (vi) Be familiar with basic database storage structures and access techniques: file and page organizations, indexing methods including Btree, and hashing; (vii) Master the basics of query evaluation techniques and and query optimization; (viii) Be familiar with the basic issues of transaction processing and concurrency control; and (ix) Master working successfully on a team by design and development of a database application system as part of a team.

Course Content Concepts and methods in the definition of management of databases; Architecture of a DBMS; Data Models: relational, hierarchical and network models; Relations, attributes, domains, etc.; Data manipulation languages; Programming in a database environment: SQL, ORACLE; Topics in database security, integrity, recovery, and concurrence; File organizations: sequential, random, indexed sequential, hierarchical, heap, has-addressed, inverted.; Database administration; Distributed database systems.

References

- Database Systems, by Hellerstein & Stonebraker, MIT Press, 4th Edition, 2004.

6.7.4 BSE 3102: Object-Oriented Software Engineering (4 CU)

Course Objectives Upon completion of the course, students will be able to: (i) Understand the entire software engineering project process, which consists of object-oriented analysis, design, programming and testing; (ii) Understand basic object-oriented programming concepts; (iii) Effectively use the main features of the object-oriented programming language Java; (iv) Gain experience in implementing object-oriented programs in Java; (v) Apply an iterative, use case-driven process to the development of a robust design model; (vi) Use UML to represent the design model; (vii) Apply the OO concepts abstraction, encapsulation, inheritance, hierarchy, modularity, and polymorphism to the development of a robust design model; and (viii) Design and implement a software system using object-oriented software engineering paradigm.

Course Content In particular the course covers the following main themes and associated topics: (i)Systems Modelling and Design Software Engineering processes and principles of good software design. UML modelling and design with particular emphasis on software architecture, behavioural modelling, object interactions and state-charts. (ii)Formal specifications of system requirements Principles of formal specifications and object orientation, state and operation schemas; class schemas, object aggregation and inheritance; specification of dependency and information sharing; definition of class union; semantic issues; reasoning techniques for validating invariant properties, such as safety and liveness. (iii)Specification of object oriented programs Principles of object-oriented program specifications, concepts of mid-conditions, pre-conditions and post-conditions of methods, the role and definition of class invariants; loop invariants as a programming technique; techniques for reasoning about the correctness of programs. (iv)The development process Generation of specifications of object-oriented programs from formal specifications of system requirement; validation and verification processes of software system; acquisition of practical experience in modelling and specifying Java programs, using appropriate tools for developing and checking specifications of Java programs.

References

- Larman C 2002, Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process, Second Edition, Prentice-Hall
- Schach, S. R., Object Oriented and Classical Software Engineering, 7th edition, McGraw-Hill, 2007
- Pressman R. S. and Ince D., Software Engineering A Practitioners Approach, McGraw-Hill, 2007
- Sommerville, I., Software Engineering, 8th edition, Addison Wesley, 2006
- Thayer, R.H. and Christiansen, M. J., Software Engineering, Volume 1: The Development Process, 3rd edition, Wiley & Sons, 2005
- Thayer, R. H. and Dorfman, M., Software Engineering, Volume 2: The Supporting Processes, 3rd edition, Wiley & Sons, 2005

6.7.5 BSE 3107: Real-Time and Embedded Systems (4 CU)

Course objectives: At the end of the couse students should be able to: (i) Demonstrate knowledge of rate monotonic theory and how to apply it to real-time applications; (ii) Understand the process and fundamentals of integrating microprocessor-based embedded system elements to realize systems that not only meet functional requirements, but timing and performance requirements as well; and (iii) Use practical skills to design and integrate a real-time operation system with a microprocessor to host real-time service data processing.

Course content: Rate Monotonic Theory, specifically the RM least upper bound, necessary and sufficient feasibility tests, and application of rate monotonic theory , Real-time Operating Systems, scheduling, synchronization mechanisms, and resource management, Embedded system architectures: System-on-chip, scalable bus architectures, memory subsystems, Sensor and Actuator IO: ADC, DAC, servos, relays, stepper motors, H-bridge, and CODECs, Real-time embedded test equipment, software debug tools, and methods of performance profiling and tracing, Real-time applications including voice/packet-switched links and networks, streaming video media, computer vision, digital control, and robotic system command and control

References

- M. Ben-Ari, Principles of Concurrent and Distributed Programming, Prentice Hall, 1990
- Stuart Bennett. Real-Time Computer Control: An Introduction, Printice Hall International, Series in Systems and Control Engineering, 1988
- Alan Burns and Andy Wellings, Real-Time Systems and Programming Languages, Addison Wesley, third edition, 2001

6.7.6 BSE 3103: Requirements Engineering (4 CU)

Course objectives: At the end of the couse students should be able to: (i) Understand the principles, tools, and techniques for requirements elicitation, specification, and analysis; (ii) Demonstrate the role of requirements in system development and maintenance; and (iii) Appreciate the difficulties of specifying requirements for real systems, as well as effective methods tools and techniques.

Course Content System and Software System Engineering, Software Requirements Concepts, Requirements Elicitation, Software Requirements Analysis, Software Requirements Specifications, Software Requirements Tools, Software Requirements Verification, Software Requirements Engineering Management, Developing a Successful Software Requirement.

References

- System Requirements Engineering, P. Loucopoulos and V. Karakostas, McGraw-Hill
- Managing Software Requirements: A Use Case Approach, 2nd edition, Dean Leffingwell, Don Widrig, Addison Wesley: Boston
- Non-Functional Requirements in Software Engineering, L. Chung, B. Nixon, E. Yu and J. Mylopoulos, Kluwer Academic Publishing, 2000
- Software Requirements: Objects, Functions, & States, A. M. Davis, Prentice Hall: Englewood Cliffs, 1993.
- System and Software Requirements Engineering: Tutorial, R. H. Thayer and M. Dortman (Editors), IEEE Computer Society Press

6.7.7 BSE 3105: Software Evolution (4 CU)

Course Objectives In this course, students will learn: (i) How selected software systems can be analyzed to understand properties of their evolution; and (ii) Interpret their implication.

Course Content Topics include: Separate compilation; design issues; verification and validation; integrating components; documentation , Issues in object-oriented programming; parallelism; event-centered programming; common design patterns; software reuse, The Laws of Software Evolution Client-server computing: Software support needed for client and server implementation; varieties of server structures; Reverse engineering and reverse engineering tools, recognizing software architecture and design patterns in existing software systems Software transformation, migration, and reengineering, recovering software components for reuse

References

- Software Evolution: A Software Maintenance Challenge, by Lowell Jay Arthur, John Wiley & Sons, 1988, ASIN: 0471628719.
- Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices by Robert C. Seacord, Daniel Plakosh, and Grace A. Lewis, Addison-Wesley Pub Co; 1st edition, 2003, ISBN: 0321118847.
- Practical Software Maintenance: Best Practices for Managing Your Software Investment by Thomas M. Pigoski, John Wiley & Sons, 1st edition, 1996, ISBN: 0471170011.
- Designing Maintainable Software, by Dennis D. Smith, Springer-Verlag, 1999, ISBN 0387987835.
- Software Metrics: Establishing a Company-wide Program, by Robert B. Grady Prentice Hall; 1st edition, 1987, ISBN 0138218447.

6.7.8 BSE 3106: Mobile Networks and Computing (4 CU)

Course Objectives At the end of the course, students will be able to: (i) Understand the basic principle, architecture and challenges of wireless networks and mobile computing systems; and (ii)Demonstrate knowledge of software development related to mobile computing systems

Course Content The topics for wireless networking include GSM, Wireless LANs, GPRS, Mobile IP, wireless ATM, Mobile Ad Hoc Networking and internetworking with TCP/IP. As well as mobile computing models in general, the SMS Bluetooth, WAP and I-mode will be introduced as typical mobile computing systems.

References

- D. Milojicic, F. Douglis and R. Wheeler, editors, Mobility: processes, computers, and agents. Addison Wesley, 1999.

6.8 Year 3 Semester 2

6.8.1 BSE 3201: Software Architecture(4 CU)

Course Objectives At the end of the course, students will: (i) Be familiar with the latest state-of-the-art software architecture; (ii) Appreciate software system design; and (iii) Understand how system's components are meant to interact with each other.

Course Content The topics to be covered in this course unit are: Architectural styles, Components of architectural design, Connectors, components, composition, Architectural design guidance and Tools for architectural design, Achieving quality goals with architectural styles, Formal models and specifications, Analyzing software architecture with SAAM, Architecture description languages (ADLs), Architecture-based development, Patterns in software architecture, Reusing software architecture.

References

- Software Architecture in Practice, by Len Bass, Paul Clements and Rick Kazman, Addison Wesley 1998.
- Software Architecture: Perspectives on an Emerging Discipline, by Shaw and Garlan, Prentice Hall 1996.
- "UML Components: A simple process for specifying component-based software", by Cheesman and Daniels, Addison-Wesley 2000.

6.8.2 BSE 3202: Distributed Systems Development (4 CU)

Course Objectives At the end of the couse students should be able to: (i) Present a conceptual model of distributed systems; (ii) Describe key components of a distributed system and evaluate the tradeoffs of alternative architectural models; (iii) Suggest algorithm suitable for application in distributed systems; (iv) Build prototype implementations of distributed systems; and (v)Demonstrate an understanding of the challenges faced by future distributed systems

Course Content Topics covered include event-driven software architectures, distributed object computing, and developing, documenting, and testing applications using object-oriented frameworks and design patterns. Techniques that enable the construction of reusable, extensible, ef- ficient, and maintainable concurrent and distributed software systems are emphasized. Abstraction based on patterns and object-oriented techniques will be crucial throughout the course, and their application studied in several in-depth case studies.

References

- S. Tanenbaum and M. V. Steen, Distributed Systems: Principles and Paradigms, Second Edition, Prentice Hall, 2006, ISBN: 0132392275.Reference Books:
- G. Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems: Concepts and Design, 3rd Edition, Addison-Wesley, 2000, ISBN: 0201619180.
- R. Anderson, Security Engineering: A Guide to Building Dependable Distributed Systems, John Wiley & Sons, 2001, ISBN: 0471389226.

6.8.3 BSE 3205: UNIX Shell Programming (4 CU)

Course Objectives At the end of the couse students should be able to: (i) use the unix operating system for information handling and software development; and (ii) develop engineering applications using c lanaguage in a unix environment.

Course Content This course introduces the UNIX operating system as a basic environment for information handling and software development. It covers the UNIX file system, job control, and processes using both the C shell and Korn shell, common UNIX utilities (sed, awk, grep), UNIX networking utilities, piping and redirection, and an introduction to shell programming.

Computer engineering applications programming using the C language in a UNIX environment. Use of UNIX tools including filters and shell scripts. Overview of UNIX software design practices using tools such as Make and SCCS. The UNIX system interface. Software design projects.

Other topics include advanced programming in both the C shell and the Korn shell using project management tools (SCCS or RCS), using software development tools (make), and an introduction to system administration.

References

- The New KornShell Command And Programming Language, by Morris I. Bolsky, David G. Korn (Contributor).
- Korn Shell Programming by Example, by Dennis O'Brien, David Pitts (Contributor).
- The Korn Shell Linux and Unix Programming Manual (2nd Edn) by Anatole Olczak.
- Linux Shell Scripting with Bash by Ken O. Burtch.
- Unix Shell Programming by Stephen Kochan and Patrick Wood (third Edition).

6.8.4 BSE 3206: Computer Graphics (4 CU)

Course objectives: Upon successful completion of this course students should be able to: (i) Demonstrate knowledge of a general purpose graphics system and its use; (ii) Show that consistent design of user interfaces based on existing standards are important; and (iii) Appreciate the domain of computer graphics and graphical user interfaces in general.

Course content: The course includes: graphics hardware, geometrical transformations, surface and volume visualisation, design and implementation of graphical user interfaces. Two dimensional imaging processes. Computer graphics applications. Introduction to computer graphics; Display system organization; Display devices and modes; Display file construction and its structure; Graphic primitive - device initialization, view porting and windowing; Line drawing, simple and symmetrical Digital Di.erential Analysis (DDA); Arch and circle generating DDA Line; and polygon clipping algorithms; Curve plotting; Transformations- projections and perspective views; Picture segmentation: Graphics standards - PHIGS and GKS.

References

- Computer Graphics (2nd Edition C Version) - Hearn, D., Baker, M.P. Prentice-Hall, 1997 (Entry Level book, good Illustrations): s.n.2176244-2nd ed. C version 1997, s.n. 2156999-2nd ed. 1994, s.n. 2033430-1986.
- Computer Graphics - Principles and Practice (2nd Edition in C). J.D.Foley, A.Van-Dam, S.K.Feiner and J.F.Hughes. Addison-Wesley, 1996. (High level book, good in depth cover of many course aspects.): s.n.2186305-2nd ed. in C 1996, s.n.2092107-2nd ed.1990 .
- Interactive Computer Graphics - Functional, Procedural and Device-Level Methods. P.Burger and D.Gillies. Add ison-Wesley, 1989. (Little correspondance with course, old fashioned.) s.n.2096479.
- Advanced Animation and Rendering Techniques. A.Watt and M.Watt. Addison-Wesley, 1992. (Extended material, for further reading and special interests.) s.n.2141145.
- Programming Windows with MFC. Jeff Prosise, Microsoft Press; 2nd edition, 1999. (The book on MFC).
- OpenGL Programming Guide. Woo, Manson., et al. Addison-Wesley, 1999 (official OpenGL red-book). s.n.2210662 -3rd ed. 1999, s.n. 2141236-1st ed. 1993.

6.8.5 BSE 3203: Object-Oriented Programming (4 CU)

Course Objective In this course, students will learn: (i) Object oriented design and software development by performing and discussing OO design for re-use of general purpose applications and small Java applets; (ii) Demonstrate correct use of the basic Java features in a working program: objects, classes, methods, IO handling, decisions and iterations; (iii) Demonstrate correct use of the following advanced Java features in a working program: inheritance, encapsulation, overloading, polymorphism, abstract classes and interfaces; and (iv) Demonstrate knowledge of GUI-based event-driven programming in a working program assignment utilizing Java GUI components, event-listeners and event-handlers.

Content: Topics include the Java API and Abstract Windowing Toolkit. Other topics covered include the use of Java as an object-oriented programming language including encapsulation, simple inheritance, and polymorphism; design of Java classes using Java interfaces and packages; implement design patterns in working Java code, and demonstrate use of Java Base Classes, including AWT. Java foundation classes including Swing and JavaBeans will be discussed briefly, along with the selection and application of current design and development tools.

References

- John Hubbard, Programming in C++ McGraw Hill Schaum's Outline Series, 2000.
- Y. Daniel Liang, Introduction to Programming with C++ (Brief Version), Prentice Hall, 2007.

6.8.6 CSC 3202: Concepts of CAD/CAM (4 CU)

Course Objective Upon successful completition students will be able to: (i) Understand the basic concepts of CAD/CAM; (ii) Demonstrate knowledge and skill in applying computer and scientific principles related to solving engineering and manufacturing problems; and (iii) Develop an understanding of engineering design and learn how to provide technical assistance in managing CAD/CAM systems

Content: Topics include: Concepts of modelling in CAD/CAM; Designing of specifications for interactive software packages in grafting; Designing and manufacturing applications; Design and development of algorithms; Methodologies of problemsolving in design; Role of geometric models in CAD/CAM; Techniques of displaying models using computer optimization; Concepts of data representations in CAD/CAM problems; Discussion on large numbers of case studies involving software development of various problems in CAD/CAM.

References

- Ibrahim Zeid, CAD/CAM, Tata McGraw Hill, New Delhi.
- J. Rooney & P.Steadman, Principles of Computer Aided Design, Pitman/ Open University, London.
- Joe Rooney & Philip Steadman, Computer Aided Design, Pitman/Open University, London.
- Glen Mallineuse, Computational Concepts and Methods, Kogan Page Ltd. London.
- Daniel L. Rayan, Computer Aided Graphical Design, Marcel Dekker, New York.

6.8.7 BSE 3204: Human-Computer Interface Design (4 CU)

Course Objective The students will throughout the course: (i) Get a thorough understanding of the nature of man-machine communication; (ii) Use acquired knowledge in designing user interfaces; (iii) Become become familiar with the variety of design and evaluation methods used in interface design; and (iv) Use these methods in their projects.

Content: Methods and techniques for designing user interfaces will be presented. These will include command-based systems, menus, graphical interfaces, desktop techniques and direct manipulation. Principles for user-friendly interfaces will be discussed using case studies from commercial programs (Windows, Word, DOS, UNIX, LINUX, others).

Human performance: memory, perception, skill learning, task closures motivation, performance, motor skills, attention; Types of users: casual, naive, learning, expert; Collecting and evaluating behavioral data; System performance and its e.ect on users: response data; Styles of interaction: command language, form-fill, direct manipulation, hierarchical/network systems; desk-top metaphor, browsing; Help mechanisms: conventional, contextsensitive, hyper text, error handling, implicit cues; Graphics: tools and graphical user interfaces; Dialogue design methods: dialogue principles, dialogue description languages; task analysis and ergonomics.

References

- Designing the User Interface: Strategies for Effective Human-Computer Interaction, 4th Edition By Ben Shneiderman & Catherine Plaisant, Publisher: Addison Wesley.

6.9 Year 3 Recess Term

6.9.1 BSE 3301: Internship (4 CU)

Course Objectives Upon successful completion students will: (i) Have gained work experience that allows them to sample professional environments in which they might seek careers; and (ii) Have experience that will help prepare them for careers and research.

Course Content The students shall be attached to an industrial partner with business in the area of software engineering or heavy bias in Information technology. During this time, the student shall be visited atleast twice by a professor or lecturer to assess the performance of the student in the industry.

The students shall acquire hands on training in maintenance, servicing and troubleshooting, and training in computer skills and applications as well as software development. Hardware maintenance; Practicals; On Job training;project management, requirement engineering, product conceptualized, Report writing including communication skills, project coordination, team organisation, product branding.

Based on the trading experience the candidate will be expected to pick ideas to develop for the final year project.

References There is no particular reference material for internship, except whatever may be recommended by the student supervisors

6.10 Year 4 Semester 1

6.10.1 BSE 4100: Software Engineering project I (5 CU)

Course Objectives Upon successful completion of this course the student will have ability to: (i) Demonstrate independent skills in collecting requirements, documenting user requirements and technical requirements for non-trivial software engineering/ research projects by pursuing a lengthy Software engineering project; and (ii) Demonstrate skills of specifying, designing and implementing a project, with assistance of one of the Professors/ Lecturers as adviser/ supervisor.

Course Content The student develops a framework within which research will be conducted and offers evidence of qualifications to pursue the research. Concepts and theories underlying the student's Project research are articulated, the problem is clearly stated, and specific, measurable goals are specified, a literature review is presented, the methods of conducting research are delineated, and strategy to achieve the goal is given.

The specific deliverables are (i) a requirements document clearly specified in well-known and acceptable notaton (ii) Technical Specification Document (iii) Detailed design document.

References No particular reference will be used for this course unit,except whatever is recommended by the student supervisors.

6.10.2 BSE 4101: Software Reliability and Testing (4 CU)

Course Objectives Upon successful completion of this course the student will have ability to: (i) Understand the software reliability process and reliability growth models; (ii) Show techniques to improve and predict software reliability; and (iii) Appreciate concepts such as operational profiles, techniques to improve and predict software reliability, preparing and executing test, black box testing, white box testing, unit testing, system testing, and integration testing.

Course content: This course introduces software reliability process, reliability growth models and shows techniques to improve and predict software reliability. Concepts such as defining necessary

reliability, developing operational profiles, techniques to improve and predict software reliability, preparing and executing test, black box testing, white box testing, unit testing, system testing, and integration testing will be explained.

References

- Software Reliability Engineering: More Reliable Software Faster and Cheaper, John D. Musa, (632 p.), Authorhouse, 2nd edition, 2004.ISBN 1418493872.
- Handbook of Software Reliability Engineering, Michael R. Lyu (Editor), McGraw Hill (1996).ISBN: 0-07-039400-8.
- Software Reliability: Measurement, Prediction and Application, J.D. Musa, A. Iannini, K. Okumoto, (621 p.), McGraw-Hill (1987).ISBN 0-07-044093-X.
- Effective Methods for Software Testing, William E. Perry, 2nd edition, John Wiley and Sons (2000). ISBN: 0-471-35418-X.

6.10.3 BSE 4103: Entrepreneurship and Business (3 CU)

Course Objectives Upon successful completion of this course the student will have ability to: (i) Systematically apply an entrepreneurial way of thinking that will allow them to create and identify business opportunities that may be commercialised successfully; and (ii) Appreciate the differences between newly formed ventures, existing small to medium size growth-oriented ventures, and entrepreneurial ventures within larger organisations.

Course Content Entrepreneurship is the process of building something from nothing. It is the process of seizing or creating opportunity without regard to the resources you control. The organizational aspects of consultancies and client responsibilities within the framework of corporate management. Topics include: contracts fees, legal issues, patents, copyrights, hourly rates, hiring and employment concerns, timekeeping and invoicing, and intellectual property. The course involves preparation of proposals and visits by appropriate professionals

References

- James W. Halloran, The McGraw-Hill 36-hour Course: Entrepreneurship, McGraw-Hill 1994
- Bruce R. Barrington, One Key Course Compass, Student Access Kit to Entrepreneurship, Prentice Hall (2005)

6.10.4 BSE 4102: Ethics for Professional Engineers (3CU)

course objectives Upon successful completion of this course, the student will: (i) Apply the ethical concepts relevant to resolving moral issues in business, industry, and other relevant areas of concern; and (ii) Articulate and defend with good reasons his/her own ethical point of view pertaining to specific problem areas in business, industry, and related areas.

course content This course examines and evaluates the meaning of ethics and professional conduct. A guiding theme is the human search or quest for values and ethical direction in terms of professional conduct and our daily life relationships with others. Students are expected to articulate and evaluate their own ethical principles and values and their foundations. The first part of this course covers the nature of ethics, ethical development, responsibilities and basic ethical directions such as Aristotelian ethics, utilitarian ethics, Kantian ethics and rights, and various views of justice. The second part of the course covers specific business and engineering ethical issues such as the company's and engineer's ethical obligation to the public, employer-employee ethical obligations including such topics as the giving and receiving of gifts, employee theft, trade secrets, computer ethics, fair wages, safety, working conditions, job satisfaction, employee rights with special emphasis

on whistle-blowing, the ethics of political tactics to advance one's career, and discrimination and affirmative action. Also, emphasis is given to environmental ethics including such topics as pollution control, the conservation of natural resources, various ethical positions on the environment, as well as such topics as biomedical ethics, treatment of animals, and the ethical assessment of new technologies.

References

- Charles B. Fleddermann, Engineering Ethics, 1st edition Prentice Hall 1999. ISBN 13: 9780137842247
- Kemper, John D.; Sanders, Billy R., Engineers and Their Profession, Oxford University Press, USA ISBN-13: 9780195120578
- Moodley, Krisen, Engineering, Business and Professional Ethics, Elsevier Science & Technology 2007, ISBN-13: 9780750667418

6.11 Year 4 Semester 2

6.11.1 BSE 4200: Software Engineering project II (5 CU)

Course Objectives Upon successful completion of this course the student will have ability to: (i) Demonstrate independent skills in implementing non-trivial software engineering/ research projects by pursuing a lengthy Software engineering project; and (ii) Demonstrate skills of Documenting, deploying a testing a well engineered solution, with assistance of one of the Professors/ Lecturers as adviser/ supervisor.

Course Content The student implements, documents, tests and deploys a software solution using the state of the art principles, concepts and technologies

The specific deliverables are: (i) Software Implementation using state of the art technologies (ii) Detailed software documentation in accordance to well known practices (iii) Installation Manuals (iv) User Manual (v) Testing and validation strategy.

References No particular reference will be used for this course unit,except whatever is recommended by the student supervisors.

6.11.2 BSE 4201: Software Design Patterns (4 CU)

Course Objectives Design patterns are standard solutions to common software design problems. Instead of focusing on how individual components work, design patterns are a systematic approach that focus and describe abstract systems of interaction between classes, objects, and communication flow. This course explores advanced principles of object-oriented design by studying key software design patterns. The patterns are drawn from a variety of sources and illustrated through examples and case studies. Examples are presented in either Java, C++ or C sharp. Students will also have an opportunity to apply these patterns through a series of hands-on exercises.

Course Content

- Basic Patterns: Structural patterns Creational patterns
- Behavioral Grasp and Cooper, Vlissides Pattern Hatching
- Relationships between Patterns: Pree's Metapatterns Zimmers Relations. Tichys classification, Classification in
- Automation of Patterns: Automation of Design Patterns Together, OpenJava and Design Patterns, Compost/Recoder Refactoring Copliens Symmetries

- Historic Roots: The timeless way of building
- Advanced Patterns: Parallelism Patterns Exclusion, State Dependence, PLOP2, Coordination, Reactive patterns Analysis Patterns Re engineering Patterns Automation of Design Patterns Process Patterns Organizational Patterns
- Applications of Design Patterns: Extreme Programming Cope: Multi-Paradigm Design

references

- Erich Gamma et al. Design Patterns
- Wolfgang Pree. Design Patterns for Object-Oriented Software Development
- Frank Buschmann, Kevlin Henney, Douglas C. Schmidt "On Patterns and Pattern Languages"

6.11.3 BSE 4203: Information Technology and Society (3 CU)

Course Objectives Upon successful completion of this course, the student will: (i) Understand the impact of computerization on society; and (ii) Appreciate the variety of computerized systems in our everyday life.

Course Content Impact of computers on society; Social structures, their boundaries and development; Ethical issues (Basic honesty, sincerity, etc.) relating to system development and its use; Systems security and protection; Dependency constraints, self-reliance; Legal issues relating to computerized systems; Uses and applications of computers for national development - communication, agriculture, health care delivery, banking, administration, accounting, planning and decision making.

References

- W H Dutton, Information and Communication Technologies: Visions and Realities, Oxford University Press, 1996
- R Kling (Ed), Computerization and Controversy: Value conflicts and social choices, Academic Press, 1996
- W H Dutton, Society on the Line: information politics in the digital age, Oxford University Press, 1999
- G Walsham, Interpreting Information Systems in Organisations, John Wiley, 1993; F Webster, Theories of the Information Society, Routledge, 1995.

6.11.4 BSE 4204: Emmerging Trends in Software Engineering (4 CU)

Course Objectives Given the rapid nature of the computing descipline, the objective of this course is to equip students with a set of emmerging trends in software engineering that may be encountered in their practice.

Course Content New trends in software engineering not yet introduced in the syllabus will be covered. Current topics in software engineering and its applications.

references No particular reference will be used for this course unit,except whatever is recommended by head of department in consultation with members of the department.

Appendix A: BUDGET

For the day programme we shall have 200 students of which, 180 shall be under private sponsorship and 20 Government.

The evening programme shall be strictly private, and shall consist of 200 private students.

Details of the Budget are given below: -

INCOME

Nature of Programme	**Degree**	**No. of Students**	**Tuition/year/Stud**	**Total Tuition per Year**
Day-Private	B.Sc.SE.	180	2,600,000	468,000,000
Day-Govt.	B.Sc.SE.	20	–	–
Evening-Private	B.Sc.SE.	200	2,600,000	520,000,000
Total Income:				988,000,000

51% of 468,000,000	238,680,000
59% of 520,000,000	306,800,000
Income to CIT:	545,480,000

EXPENDITURE

TYPE	**DETAILS**	**AMOUNT**
1) Staff Renumeration:	Lectures' Evening Teaching Allowances	120,000,000
	Visiting Professors' Allowances	48,000,000
2) Capital Development:	Computers	125,000,000
	Heavy Duty Printers	2,000,000
3) Course Materials:	Textbooks	20,000,000
	Software	30,000,000
	Backup Media	10,000,000
4) Stationary:	Photocopier Toner	2,000,000
	Photocopying Paper	6,000,000
	Other Office Requirements	10,000,000
5) Administration:	Administrative Staff Allowances	20,000,000
6) Maintenance of equipment		20,000,000
7) Research and student projects		50,000,000
8) Capital Development		20,000,000
9) Contract staff salaries		40,000,000
9) Academic Functions:	Workshops/ Seminars	10,000,000
	International Conferences	10,000,000
Total Expenditure		543,000,000
Surplus:	Income to CIT - Total Expenditure	2,480,000

BUDGET DETAILS

Number of course units per semester	40
Number of contact (lecture) weeks per semester	15
Number of contact hours per week per course unit	4
Rate per contact hour	50,000
Cost per Computer	2,500,000
Number of Computers	50
Visiting Professors' Allowance per semester per Professor	24,000,000
Number of visiting Professors per semester	2

Note: All the money quotations in the budget are in Uganda Shillings.

Appendix B: ACADEMIC STAFF LIST

	NAME	HIGHEST DEGREE	RANK
	Department of Networks:		
1	Dr. Fisseha Mekuria	PhD	Visiting Professor
2	Dr. Idris A. Rai	PhD	Associate Professor
3	Dr.Benjamin Kanagwa	PhD	Lecturer
4	Dr. Tonny Bulenga	PhD	Lecturer
5	Paul Bagyenda	M.Phil	Lecturer
6	Julianne Sansa Otim	M.Sc.	Assistant Lecturer
7	Agnes F Namulindwa	M.Sc.	Assistant Lecturer
8	Mariam Sensalire	M.Sc	Assistant Lecturer
9	Fred N. Kiwanuka	M.Sc.	Assistant Lecturer
10	James N Kasigwa	M.Sc	Assistant Lecturer
11	Charity B Mulenga	M.Sc.	Assistant Lecturer
12	Drake Patrick Mirembe	M.Sc.	Teaching Assistant
13	Brian Joseph Mukwaya	B.Sc.	Teaching Assistant
	Department of Computer Science:		
1	Prof. Venansius Baryamureeba	PhD	Professor
2	Dr. Jose Ghislain Quenum	PhD	Senior Lecturer
3	John Quinn	PhD	Lecturer
4	John Ngubiri	PhD	Assistant Lecturer
5	Florence Tushabe	M.Sc.	Assistant Lecturer
6	NT Rwangoga	M.Sc.	Assistant Lecturer
7	Ronald Azairwe	M.Sc.	Assistant Lecturer
8	Richard Ssekibule	M.Sc.	Assistant Lecturer
9	John Kizito	M.Sc.	Teaching Assistant
10	Doreen Tuheirwe	B.Sc.	Teaching Assistant
11	Rose Nakibuule	B.Sc.	Teaching Assistant
12	Engineer Bainomugisha	M.Sc.	Teaching Assistant
13	Ezra J Agaba	B.Sc.	Teaching Assistant
14	Fred Byamugisha	M.Sc.	Assistant Lecturer(P/T)
	Department of Information Systems:		
1	Dr. Ogao Patrick Job	PhD	Associate Professor
2	Martin Bagaya	PhD	Lecturer
3	Agnes R Semwanga	M.Sc.	Assistant Lecturer
4	Josephine Nabukenya	M.Sc.	Assistant Lecturer
5	Richard Mayanja	M.Sc.	Assistant Lecturer
6	Michael Niyitegeka	MBA	Assistant Lecturer
7	Ernest Mwebaze	M.Sc.	Assistant Lecturer
8	Rehema Baguma	M.Sc.	Assistant Lecturer
9	Emily Bagarukayo	M.Sc.	Assistant Lecturer
10	Agnes Nakakawa	M.Sc.	Teaching Assistant
11	Irene Nakiyimba	M.Sc.	Teaching Assistant
12	Fiona Catherine Ssozi	B.Sc.	Teaching Assistant
13	Peter Khisa Wakholi	B.Sc.	Teaching Assistant
14	Mercy Amiyo	B.Sc.	Teaching Assistant
15	Noreda Kiremire	M.Sc.	Assistant Lecturer(P/T)
16	Abdul Nsubuga	M.Sc.	Assistant Lecturer(P/T)

	NAME	HIGHEST DEGREE	RANK
	Department of Information Technology:		
1	Irina Ya. Zlotnikova	PhD	Visiting Professor
2	Jude T Lubega	PhD	Lecturer
3	Joseph M Ssemwogerere	M.Sc.	Lecturer
4	Raymond Mugwanya	M.Sc.	Assistant Lecturer
5	Gilbert Maiga	M.Sc.	Assistant Lecturer
6	Moses Niwe	M.Sc.	Teaching Assistant
7	Fiona Tulinayo	M.Sc.	Assistant Lecturer
8	Aminah Zawedde	M.Sc.	Assistant Lecturer
9	Margaret Nagwovuma	B.Sc.	Teaching Assistant
10	Emmanuel Acuc	M.Sc.	Assistant Lecturer(P/T)
11	Consolata Bitature Mugambe	M.Sc.	Assistant Lecturer(P/T)
12	Daudi Jingo	M.Sc.	Assistant Lecturer(P/T)
13	Bizimungu Omara	M.Sc.	Assistant Lecturer(P/T)
14	Faridah Kyambadde	M.Sc.	Assistant Lecturer(P/T)
15	Jackson Muhirwe	M.Sc.	Assistant Lecturer(P/T)
16	Martin Ngobye	M.Sc.	Assistant Lecturer(P/T)
17	Martin Gordon Mubangizi	M.Sc.	Assistant Lecturer(P/T)
18	Abudu Nasser Ntege	M.Sc.	Assistant Lecturer(P/T)
19	Asaph Tweheyo	MBA	Assistant Lecturer(P/T)
20	Sylvia Sanyu Mukwaya	M.Sc.	Assistant Lecturer(P/T)
21	Annette Kebba Sengendo	M.Sc.	Assistant Lecturer(P/T)
	Part-time		
1	Mango, J.M.	PhD	Senior Lecturer
2	V Ssembatya	PhD	Senior Lecturer
3	Katiti, E.B.	PhD	Senior Lecturer
4	Barongo, M.B.	M.Phil	Lecturer
5	Faculty of Arts		
6	Department of Physics		
7	Department of Electrical Engineering		
8	Department of Mathematics		

END.