Subject Description Form

Subject Code	EE2007C
Subject Title	Computer System Fundamentals
Credit Value	3
Level	2
Pre-requisite/ Co-requisite/ Exclusion	Nil
Objectives	1. To enable students to establish a broad knowledge of the structure of a generic computer system. 2. To enable students to understand software development in an embedded system. 3. To enable students to utilize a microcontroller to solve engineering problems
Intended Learning Outcomes	Upon completion of the subject, students will be able to: a. Given specifications of an application, design the software to carry out the necessary operations in an embedded system. b. Understand advanced features of the latest microprocessors and understand functions of basic computer peripherals. c. Given a set of conditions, design a computer system to solve the problem d. Think logically and be able to present results in writing.
Subject Synopsis/ Indicative Syllabus	**Computer Systems Hardware and Operations** 1. *Microprocessor operation and internal architecture:* Operations of registers, buses and data path, operations of ALU and general pipeline architecture. Introduction to structure and operation of a modern microprocessor. 2. *Memory organization:* Characteristics of current memory technologies. Memory hierarchies and memory decoding mechanism. 3. *Input and output systems:* Direct I/O system and memory mapped I/O; I2C and SPI; interrupt and polling mechanisms. 4. *Microprocessor hardware and interfacing:* System bus organization and interfacing techniques, CPU bus timing, system bus structure. **Introduction to Embedded System and Software** 1. *Introduction to Arduino software tool:* Integrated Development Environment (IDE) for Arduino, basic program syntax. 2. *Input/Output operations and applications:* Introduction to serial communication, digital I/O, analogue I/O and PWM (Pulse Width Modulation). 3. Introduction to assembly language programming **Laboratory Experiment:** Perform basic input/output operations of an embedded computer systems by C programming language with the Arduino processor Speed control of a DC motor based on input from a sensor Implement the interrupt service program in an embedded microcontroller.

Teaching/ Learning Methodology

Lectures and tutorials are the primary means of conveying the basic concepts and theories. Experiences on design, practical applications and programming are given through experiments, in which the students are expected to solve design problems with real-life constraints and to attain feasible solutions with critical and analytical thinking. Interactive laboratory sessions are introduced to encourage better preparation and hence understanding of the experiments. On-the-spot assessments are conducted in the laboratory to provide additional incentives for student learning. Experiments are designed to supplement the lecturing materials, especially in software development, so that students are encouraged to take extra readings and to look for relevant information.

Teaching/Learning Methodology	Outcomes			
	a	b	c	d
Lectures and tutorials	✓	✓	✓	
Tutorials	✓	✓	✓	
Experiments	✓		✓	✓

Assessment Methods in Alignment with Intended Learning Outcomes

Specific assessment methods/tasks	% Weighting	Intended subject learning outcomes to be assessed			
		a	b	c	d
1. Examination	60%	✓	✓	✓	✓
2. Test	15%	✓		✓	
3. Laboratory performance & report	15%	✓			✓
4. Exercises	10%	✓	✓	✓	
Total	100%				

It is a fundamental computer architecture subject. The outcomes on concepts, design and applications are assessed by the usual means of examination and test whilst those on analytical skills, problem-solving techniques and practical considerations of programming, as well as technical reporting are evaluated by experiments, and the report.

Student Study Effort Expected

Class contact:	
▪ Lecture/ Tutorial	30 Hrs.
▪ Laboratory	9 Hrs.
Other student study effort:	
▪ Laboratory preparation/report	11 Hrs.
▪ Self-study	50 Hrs.
Total student study effort	100 Hrs.

Reading List and References	**Textbook:** 1. C. Hamacher, Z. Vranesic, S. Zaky, and N. Manjikian, Computer Organization and Embedded Systems, 6th Edition, McGraw-Hill, 2012 2. J.L. Hennessy and D.A. Patterson, Computer Architecture: A Quantitative Approach, 6th Edition, Elsevier, 2019 3. 3. A. Tanenbaum, T. Austin, Structured Computer Organization, Pearson India, 6th Edition, 2016. **References and online materials:** 1. A.K. Ray, Advanced Microprocessors & Peripherals, McGraw-Hill, 2006 2. R.J. Tocci and F.J. Ambrosio, Microprocessors and Microcomputers: Hardware and Software, 6th Edition, Prentice Hall, 2003 3. B.B. Brey, The Intel Microprocessors Architecture, Programming, and Interfacing. 8th Edition, Prentice Hall, 2008. 4. https://www.hackerearth.com/blog/developers/arduino-programming-for-beginners/

July 2022