1. **Course Title:** Principles of Cyber-Physical Systems Computation
2. **Course Code:** L-T-P-C: 3 - 0  1 - 3
3. **About this Course:**

   In this course, we will learn the requirements of the Cyber-Physical System software such as safety, timeliness, stability, and performance. These software requirements are designing, analyzing, and implementation of Cyber-Physical Systems in a principled manner. As mentioned above, the characteristic difference between the general software and the software for CPS is that the former reacts with the external environment and so the inputs/outputs of the environment, dynamic nature of the environment, the communication among the components of the systems, etc should be taken into consideration.

4. **Course Objectives:**

   To understand the varied requirements of programming CPS compared to general-purpose systems. To account for the reactive nature of CPS with the environment in contrast to general-purpose systems. To account for the dynamic nature of the CPS. To know the model-based design approach for building CPS and meeting the requirements of design such as safety, liveliness, stability, and performance.

5. **Modified Course Outcomes**

| | |
|---|---|
| 1 | Ability to identify the difference in design automation flow for the CPS and General Purpose System. |
| 2 | Ability to model a real-life case study (e.g. car cruise controller) with reactive components using the discrete, synchronous model, and asynchronous models. |
| 3 | Ability to understand and use tools adopted in industry for formal verification of CPS |

6. **Detailed Objectives:**

   Often students think Cyber-Physical Systems (CPS) are hardware related to Electronics and Communication Engineering, on the contrary, CPS is an interdisciplinary subject. The cyber part of CPS is the computation and includes writing software that reacts with the external environment. In fact, there is a Turing Award (Nobel Prize in Computer Science) given for the contributions that computer scientists made in this field of CPS. So the principles we use to develop general software are different from the principles required for writing software for Cyber-Physical Systems. The motivation for computer science students to learn this is because there are going to be numerous

opportunities for computer science students in the CPS field in the immediate future. The Government of India has created a National-Mission of Cyber-Physical Systems to address the challenges and needs in this field in the near future.

In this course, we will learn the requirements of the Cyber-Physical System software such as safety, timeliness, stability, and performance. These software requirements are designing, analyzing, and implementation of Cyber-Physical Systems in a principled manner. As mentioned above, the characteristic difference between the general software and the software for CPS is that the former reacts with the external environment and so the inputs/outputs of the environment, dynamic nature of the environment, the communication among the components of the systems, etc should be taken into consideration.

**7. Prerequisites:**

Computer Organization/Architecture, Theory of Computation

**8. High-Level Course Description (Course Topics)**

| Module | Name |
|---|---|
| 1 | Introduction |
| 2 | Synchronous Model Part 1 |
| 3 | Synchronous Model Part 2 |
| 4 | Safety Requirements |
| 5 | Asynchronous Model |
| 6 | Liveness Requirements |

## 9. Detailed Description

| Module | Name | Contents | Hours |
|---|---|---|---|
| 1 | Introduction | What is CPS? Applications of CPS. Design Process: Modeling, Design, and Analysis. | 3 |
| 2 | Synchronous Model Part 1 | Reactive Components-Variables, Valuations, and Expressions; Inputs, Outputs, and States, Initialization, Update, Executions, and Extended State Machine. Properties of Components-Finite State Components, Combinational Components, Event-Triggered Components, Nondeterministic Components, Input-Enabled Components, Task graph, and Await Dependencies. | 6 |
| 3 | Synchronous Model Part 2 | Composing Components-Block Diagram, Input and Output Variable Renaming, Parallel Composition, and Output Hiding. Synchronous Design-Synchronous Circuits, Cruise Controls, Synchronous Networks. | 6 |
| 4 | Safety Requirements | Safety specifications: Invariants of Transition System, Role of Requirements in System Design, Safety Monitors. Verifying Invariants- Proving Invariants, Automated Invariant Verification, Enumerative Search, Symbolic Search | 6 |
| 5 | Asynchronous Model | Asynchronous Processes-States, Inputs, and Outputs, Input, Output, and Internal Actions, Executions, Extended State Machines, Operations on Processes, Safety Requirements, Asynchronous Design Primitives Blocking Vs Non-Blocking Synchronization, Deadlocks, Shared Memory, Fairness Assumptions, Asynchronous Coordination Protocols | 9 |
| 6 | Liveness Requirements | Temporal Logic-Linear Temporal Logic, LTL Specifications, LTL Specifications for Asynchronous Processes and Model Checking-Buchi Automata, From LTL to Buchi Automata, Nested Depth First Search, Symbolic Repeatability Checking | 6 |
| | | Total Hours | 36 |

## 10. Text, References Books and Online Content

a. Principles of Cyber-Physical System, Rajeev Alur, MIT Press, 2015
b. CIS 540: Principles of Embedded Computation, Spring 2019

## 11. Assessment Plan

a. 3 Lab Assignments 30%: (Each Lab Assignment is 10%)
b. 2 Mini Projects: 20% (Each Project Component is 10%)

c. Mid Exam *(Open Notes Exam, but limited to two pages)*: 10% (Written component and Viva Component)
d. End exam *(Open Notes Exam, but limited to four pages)*: 30% (Written component and Viva Component)
e. Class Participation 10% (Surprise Quizzes Best 7 out of 12 Quizzes; These quizzes will be negative marking (0.25 for each wrong answer); Middle of the class or end of the class;)