AMITY UNIVERSITY
UTTAR PRADESH

**Course Title** : Aerospace Software Engineering
**Course Code** : SPAC412
**Course Level** : UG
**Credit Units** : 03

| L | T | P/S | SW/FW | TOTAL CREDIT UNITS |
|---|---|---|---|---|
| 3 | 0 | 0 | -- | 03 |

**Course Objectives:** The objectives is to: familiarize the student with the Software engineering processes , testing and verification. The methods employed to develop Mission critical software for Aerospace systems .

**Pre-requisites:**

- Aerospace Electronics., Programming in C

**Course Contents/Syllabus:**

| | Weightage (%) |
|---|---|
| **Module I Introduction** | **5%** |
| Software System definition and components, Software engineering, Types of software : mission critical software and its architecture. Hardware implications on software , Avionics software : FMS , ECAS , | |
| **Module II Software Models and Management** | **40%** |
| Software Engineering & Development process and models : Water fall , V process models , Requirement definition , design and development . software errors , error detection and correction methods, software project Management : Software Process and Metrics , Configuration Management. Computer Aided Software Engineering ( CASE) in Avionics Software development. Rapid prototyping. Examples of Mission critical software coding : modular codes , Reentrant codes and standard techniques. | |

| | |
|---|---|
| **Module III Fault Tolerant Software** | **35%** |
| Critical embedded systems , Real time systems , System of Systems, Process engineering , safety in Airborne embedded System Standard RTCA – DO 178 B, Airborne System Integration problems , Integration through Model Based Engineering , Case study of FMS System. | |
| **Module V Software Verification and Validation** | **20%** |
| Verification Process , Software Testing techniques : White Box , Black box and Basic Path Formal Methods . Test case generation methods , test Metrics , software quality control, Testing measures for Avionics and aerospace software, Software Safety and Security : its assurance of aerospace systems | |

**Student Learning Outcomes:** On completion of this course students would be able

- To understand the difference between vision critical software and general power purpose
- To know the signification of modular critical software engineering
- Write program for aerospace applications
- Analysis the software of flight control

**Pedagogy for Course Delivery:** The course pedagogy will include lectures and tutorials, The course will also include the explanation of the aerospace software systems and their design . Use of PPTs and lecture notes .Real life development cycle will be discussed.

**Assessment/ Examination Scheme:**

| Theory L/T (%) | Lab/Practical/Studio (%) | Total |
|---|---|---|
| **100%** | **0%** | **100%** |

**Theory Assessment (L&T):**

| | Continuous Assessment/Internal Assessment | | | | | End Term Examination |
|---|---|---|---|---|---|---|
| **Components (Drop down** | **ME** | **Q/S** | **CPA** | **TP** | **A** | **EE** |
| **Weightage (%)** | 10 | 5 | 5 | 5 | 5 | 70 |

***Texts and References:***

- Safety critical software development : DO 178 B standards
- Critical systems and software safty – CMU standards fillip koopman
- Fundamental of fault tolerant Computing "VP Kielson, wilay publication
- Software engineering : A Practitioner Approach" By Roger S Pressman, McGraw Hills
- " Software engineering : Tutorials" Richard Thayer , California University , Sacramento
- NASA Handbook " NASA Software Safety Guide " NASA – GB – 8719.13 NAS technical Publication.
- " Fault Tolerant Software Design " W. H. Pierce , Academic Press

*Lecture notes*

Lecture notes given in soft and hard copy from time to time.