

Introduction			Training Objectives		
<p>Duration: 1 day</p> <p>The Software Safety course will provide delegates with a general appreciation of software safety engineering issues in the context of a typical systems development life-cycle. The Software Safety Course has been developed to raise awareness of software safety issues and to promote safety culture as required by any Safety Management System.</p> <p>As well as some theoretical material, the course provides practical examples of how programmers could improve software safety throughout the software development life-cycle.</p>			<p>The aim of the Software Safety Course is to raise awareness of software safety issues and to promote safety culture as required by a Safety Management System.</p> <p>A Training Objective (TO) is a precise statement of the skills and knowledge required of a trainee at the end of a training Session. The table below details the TOs for the Software Safety course.</p>		
Session 1 – Introduction to Safety (60 min)			Session 2 – Software Systems Safety (60 min)		
TO 1.1	Need for Safety	Explain the need for formal safety management.	TO 2.1	Safety Terminology	Give a definition for some basic safety terminology including: safety, risk, error, fault, random failure, systematic failure and hazard.
TO 1.2	Functional Safety	Explain the difference between health & safety and functional safety.	TO 2.2	Safety Standards	List some of the important safety standards relevant to systems and specifically to safety-related software development.
TO 1.3	Systems and Software	Describe how software safety fits into a systems context using a representative system as an example.	TO 2.3	Safety Analysis	Describe a typical safety life-cycle and the associated system Safety Analysis methods and objectives.
TO 1.4	Safety Management	Explain how an Safety Management System is implemented and documented in a Safety Management Manual.	TO 2.4	Safety Requirements	Explain how system and software safety requirements are specified for safety-related systems.

Session 3 – Safety Requirements Exercise (60 min)			Session 5 – Programming for Safety (60 min)		
TO 3.1	Exercise	Complete an exercise to assess, review and specify system and software safety requirements for an example aircraft wheel braking system.	TO 5.1	Real-Time Issues	Identify the typical real-time software issues that can adversely impact upon the safety of a system.
Session 4 – Software Safety Assurance (90 min)			TO 5.2	Partitioning & Redundancy	Explain how partitioning and logical and physical software partitioning can maintain the integrity of a software system.
TO 4.1	Safety Management	Describe the typical contents of a Safety Management Manual and explain the safety processes and responsibilities detailed therein.	TO 5.3	Defensive Programming	Explain some popular defensive programming techniques and discuss software reuse and the use of Commercial-Off-The-Shelf (COTS) software within this context.
TO 4.2	Safety Cases	Explain what a safety case is and why it is required. Explain the scope and purpose of a Design Safety Case (DSC) and describe the relationship between a DSC and an Operational Safety Case. Explain what a Software Safety Analysis (SSA) is and explain the differences between an SSA and a DSC.	TO 5.4	Languages & Safe subsets	Compare the use of Ada, C, C++ and Java programming languages for safety-related systems and describe the purpose of safe subsets.
TO 4.3	Safety Argument	Describe the constituent parts of a safety argument and explain Goal Structuring Notation.	TO 5.5	Firmware	Explain a basic process for the safety assurance of firmware devices containing code with the potential for systematic failures.
TO 4.4	Software Safety Evidence	Describe the different types of process-based and product-based evidence required to make a software safety argument and explain how this evidence can be generated throughout the SDLC.			