

Learning outcomes describe what you should know and be able to do if you make full use of the opportunities for learning that we provide. All these skills are described in the following areas (A, B, C, D). In the individual module syllabi, the categories of learning outcomes (A, B, C, D) and the individual learning outcomes appropriate to the module are identified.

#### **A- Knowledge and Understanding of**

- A1) the system development lifecycle;
- A2) a wide range of principles and tools available to the software developer, such as software process methodologies, choice of algorithm, language, software libraries and user interface technique;
- A3) the principles of object-oriented software construction;
- A4) the software-development process, including requirements analysis, design, programming, testing and maintenance;
- A5) the range of situations in which computer systems are used, the ways in which people interact with them;
- A6) professional issues to cover: social, ethical and legal aspects;
- A7) communication issues in large, complex software projects;
- A8) the principles and techniques of a number of application areas informed by the research directions of the subject, such as software engineering, net-centric, and distributed systems.

#### **B- Intellectual (thinking) skills - able to**

- B1) model object-oriented software systems;
- B2) investigate and improve the specification of a software system;
- B3) design and plan software solutions to problems using an object-oriented strategy;
- B4) identify a range of solutions and critically evaluate and justify proposed design solutions;
- B5) write and test programs using at least one object-oriented programming language;
- B6) evaluate systems in terms of general quality attributes and possible trade-offs presented within the given problem;
- B7) use and evaluate appropriate tools and techniques
- B8) reflect and reason concerning a given information handling problem or opportunity.

#### **C- Practical skills - able to**

- C1) specify, design and construct CASE tools and application software;
- C2) use logic and discrete mathematics to specify software elements;
- C3) develop and apply testing strategies for software applications;
- C4) develop software applications in a development environment that makes use of commonly supported tools;
- C5) identify some of the main risks of software development and use;
- C6) use network information services
- C7) Prepare and deliver coherent and structured verbal and written technical reports;
- C8) use the scientific literature effectively and make discriminating use of Web resources;
- C9) analysis of system requirements and the production of system specifications;

C10) use appropriate computer-based design support tools.

**D- Transferable skills - able to**

- D1) effectively participate in team-based activities;
- D2) structure and communicate ideas effectively, both orally, in writing, and in cases involving a quantitative dimension;
- D3) use IT skills and display mature computer literacy;
- D4) work independently and with others;
- D5) manage learning and self-development, including time management and the development of organisational skills;
- D6) display personal responsibility by working to multiple deadlines in complex activities;
- D7) undertake practical training and placements in relevant organisations
- D8) appreciate the need for continuing professional development and in recognition of the need for lifelong learning

In order to provide students with the “life long learning” attitude, the teaching method is essentially based on self learning (3 hours in class rooms and 6 hours out of class rooms: coursework, practical works, workshops, seminars, etc.)