

THE ASSOCIATION OF BUSINESS PRACTITIONERS (ABP)
POSTGRADUATE DIPLOMA:
INFORMATION SYSTEMS APPLICATIONS DEVELOPMENT

Introduction

This programme is primarily designed for graduate level students who wish to achieve an entry qualification and credits towards an MSc degree in computer science. The programme will also appeal to individuals with relevant employment experience and information technology (IT) skills who need to raise their awareness of modern approaches to the development of information systems.

Additionally, the educational aims of the programme are:

- to consolidate and extend any first-degree knowledge or employment experience.
- to refresh those disciplines necessary for intensive study and personal time management.
- to acquire and enhance basic skills in research techniques and report writing sufficient for entry to postgraduate degree programmes.
- to develop knowledge in professional issues related to information technology as a means of furthering career progression prospects.

Programme Summary

The Postgraduate Diploma (PGD) can be completed in either of two streams, dependent on the individual student's prior learning:

- a. Main Stream. This alternative is the standard progression path. **All four modules**, each of 15 credits in value, must be taken:
 - (1) Introduction to Object-oriented Programming.
 - (2) Advanced Object-oriented Programming.
 - (3) Software Development Methodologies.
 - (4) Professional Issues in Information Technology.
- b. Specialist Stream. This path may be followed if a student so wishes and can also provide evidence of competence in both modules (1) and (2) above:
Modules (3) and (4), as above, must be taken with **one** of:
 - (5) Internet Applications Development (30 credits).
 - (6) Advanced Database Applications (30 credits).

Entry Requirements

Eligible candidates will be expected to satisfy one of the following requirements:

- a. A recognised and recent university degree in computer science or in a closely related subject.
- b. A recognised university degree with evidence of substantial computer literacy.
- c. Mature student with extensive responsible employment in the information and communications technology sector and a clear potential for postgraduate studies.

Method of Assessment

The programme may be internally assessed by the teaching centre and will be externally verified. Its organisation and assessment methods are summarised in the table below:

Serial	Module Title	Credits	Learning Hours				
			Lectures	Tutorials	Practical	Directed Unsupervised Learning	Total hours
1	Introduction to Object-oriented Programming.	15	30	38	45	37	150
2	Advanced Object-oriented Programming.	15	38	30	45	37	150
3	Software Development Methodologies.	15	38	30	45	37	150
4	Professional Issues in Information Technology.	15	38	30	45	37	150
5	Internet Applications Development.	30	76	60	90	74	300
6	Advanced Database Applications.	30	76	60	90	74	300

Practical Facilities

Teaching centres must be able to support an intensive high-level programme of this nature with ample practical resources. Typical facilities should include: a local area network running *Windows* or *Unix*; an object-oriented programming language such as Java or C#; a text-based and a visual development environment; an automated test utility, such as *JUnit*; a build tool such as *Apache Ant*; a choice of Internet browsers; access to an Internet server (real or virtual); a variety of scripting and mark-up languages to support the development of multi-tier web applications; a relational database such as *Oracle* or *Access*.

Main Stream Modules : (Proposed) Learning Outcomes

The broad contents of the Main Stream modules (1) to (4) are summarised below; in due course, learning outcomes and their indicative contents will be supplied for these modules.

(1) Introduction to Object-oriented Programming.

Program development; the use of text editors, compilers and interpreters.

The structure and components of a program written in a strongly-typed language.

The algorithmic approach to problem solution.

Simple data types; constants; variable declaration, assignment, expressions, statements.

The syntax and use of conditional statements, repetition, arrays, input and output statements; serial (sequential) file handling.

Program testing, error types, detection and correction; exception handling.

In-built and programmer-defined functions; parameter passing.

Detailed documentation of programs and projects.

Reference data types; instantiation of classes; class design; simple inheritance.

Concepts of access modifiers, accessor and mutator methods, abstract and final classes, encapsulation, polymorphism.

Application of programming principles to the solution of simple problems.

The development and documentation of a small non-trivial programming project.

(2) Advanced Object-oriented Programming.

The components and characteristics of the chosen programming language.

Class design; representation in the Unified Modelling Language (UML).

Alternative forms of constructor methods; overloading and overriding.

Well-formed classes; canonical forms; contracts of methods; assertions.

Single and multiple inheritance; interfaces.

Simple graphics; the design and development of graphical user interfaces.

Event handling models and related programming techniques.

Concurrency problems; thread creation and use; simple multithreaded applications.

Testing strategies; unit testing and the use of an automated testing tool.

Design patterns: model view controller, singleton, template, strategy, decorator, iterator.

Collections (data structures); appropriate choices for given applications.

The completion of a programming assignment incorporating advanced object-oriented techniques.

(3) Software Development Methodologies.

The challenges in developing complex software systems.

Alternative methodologies for software development: waterfall, prototyping, agile, the Rational Unified Process, Extreme Programming,...

Requirements capture by Use Case modelling.

Techniques for the modelling of system data.

Modelling the organisation and behaviour of a system with UML.

Productivity tools; integrated development environments.

Creating simple applications by employing a visual development environment.

The benefits of adopting design patterns.

The investigation of an emerging technology in software development.

Developing a software application based on a realistic scenario or case study using up-to-date modelling tools and techniques.

(4) Professional Issues in Information Technology.

Codes of conduct and the responsibilities of computer professionals.

Human resource aspects: the legal framework of employment, recruitment, selection, professional development.

Legal, social and ethical issues in systems projects.

The Data Protection Acts of 1984 and 1998.

The Copyright Designs and Patents Act 1988.

The Computer Misuse Act 1990.

Contracts for the supply of computer systems and services.

Information systems project management; details of the standard approaches.

The PRINCE management methodology.

International problems relating to the Internet.

The investigation of a current professional issue in information technology.

Reporting on a subject of professional interest in a standard academic format.

Specialist Stream Modules : Aims and (Proposed) Learning Outcomes

Module 5: Internet Applications Development

Aims

This module aims to enable students:

To appreciate the operation of the Internet and the technical problems in delivering distributed systems.

To acquire knowledge and skills in the implementation of distributed systems in general and of web services in particular.

To investigate and report on current trends and issues in web services.

Learning Outcomes

The operating principles of the Internet and the range of services it supports; regulatory bodies.

A working knowledge of the TCP/IP suite of protocols.

Practical web page construction using HTML, XHTML, cascading style sheets and Dynamic HTML.

Client side processing using a standard scripting language such as JavaScript; exercises involving the Document Object Model.

The role and importance of the Extensible Markup Language (XML); exercises with Document Type Definitions and the Extensible Style Sheet Language (XSL).

Technical problems and solutions relating to distributed systems; choice of implementation technologies such as .net and Java Enterprise Edition.

Revision of relational database knowledge: design of related tables and processing with SQL commands.

Server side processing by means of Java servlets, Java Server Pages, PHP, ASP.net or acceptable alternatives.

The development of a multi-tier application incorporating a SQL database, servlets (or alternatives, and client side access based on an appropriate design methodology.

Tools for the development, testing and deployment of distributed applications.

The current meaning of the term “Web Services” with examples of commercial applications such as eBay, Amazon and Google.

The underlying principles of service-oriented architectures (SOAs).

The benefits and limitations of SOAs; comparisons with alternative architectures.

Security issues with Internet applications and web services; practical countermeasures to minimise threats.

The appraisal of current trends and issues in web services, resulting in the delivery of an original research report or a working web services application.

Module 6: Advanced Database Applications

Aims

This module aims to enable students:

To obtain a sound understanding of database theory and principles.

To acquire the knowledge and skills necessary in order to participate in the design and development of relational databases and of alternative database models.

To evaluate a range of common database models in terms of their advantages, limitations and problems.

To explore and report on emerging database technologies and their applications.

Learning Outcomes

Database principles: the relational model; conceptual, logical and physical models.

Entity relationship modelling; alternative notations and practical exercises in data analysis and modelling.

The design of relational databases; representation; mapping to a relational schema; functional dependency; normalisation.

Languages for relational databases: relational algebra, relational calculus, SQL.
Development tools and their application.

Practical coursework in the design and implementation of a relational database comprising several relations.

Database management: transactions, concurrency, recovery, security; queries and their optimisation.

Distributed databases: traditional client/server systems; the design and implementation of web databases.

Comparison of relational, object-relational and object-oriented database systems.

Data warehousing; on-line analytical processing; business applications; alternative database architectures.

Data mining and its applications; knowledge discovery in databases: typical tasks and areas of interest such as visual data mining and web content mining.

Investigation, and reporting on, an emerging database technology, architecture or application.