GALWAY-MAYO INSTITUTE OF TECHNOLOGY

Institiúid Teicneolaíochta na Gaillimhe-Maigh Eo

Master of Science in Computing

(Incorporating Approved Changes – 2007)

TABLE OF CONTENTS

1. Aims and Objectives of the Masters Programme

- To provide an opportunity to new graduates and established computing professionals for personal career development by undertaking formal post-graduate education on a full-time or part-time basis.

- To provide commercial and industrial companies with a locally-based means of upgrading the computing skills and knowledge of their staff.

- To increase the number of practitioners with advanced skills and to foster a spirit of co-operation and collaborative research between GMIT and other organisations; locally, nationally and internationally.

On completion of the M.Sc the participant will:

- Have developed core skills in key areas of advanced computing techniques.
- Be suitably informed of emerging developments in these areas.
- Be able to apply best practices across a range of software-related disciplines.

2. Entry Requirements

- Honours Degree in Computing or Honours Degree in Electronics/Computer Engineering with substantial software development experience.

3. Programme Duration

- The MSc in Computing can be completed as a one year full-time programme or alternatively as a part-time programme typically over two to three years. This twelve-month programme commences in early September and concludes in late August of the following year.

- The next intake is September 2007.

- Part-time students benefit from lectures and laboratories that are scheduled in the evening time (typically 6:00PM - 9:00PM) thus allowing them to continue employment during normal business hours.

- To be eligible for the award of Master of Science (Computing), participants must successfully complete 4 taught Modules and a Masters Project. Course credits may be accumulated under the HETAC's ACCS system.

© 2007 Galway-Mayo Institute of Technology

4. Programme Content

- Each taught module consists of 100 hours class contact, with participants undertaking a substantial workload on their own.

The taught modules for the 2007 - 2008 intake are:

- Software Engineering
- Object Oriented Systems
- Advanced Databases
- Network and Forensic Computing

Full-time students must take all of the above modules, part-time students can elect to take Object Oriented Systems and / or Advanced Databases in September 2007.
The Major Project will be supervised and will be of approximately 450 - 500 hours duration.

Full-time students must also take a module in Research Methods for Computer Scientists which will develop core competencies in critical thinking, research techniques, research methodologies, technical writing & critique, and a process oriented approach to project / dissertation completion.

See ***http://gmitweb.gmit.ie/computing/courses/msc/*** for further details.

© 2007 Galway-Mayo Institute of Technology

Core Module Information and Syllabus Content

SOFTWARE ENGINEERING

MODULE CONTENT…

Objectives

On completion of this module the participant will:

- Incorporate best practice in software development
- Have a thorough understanding of the critical issues in project management, both technical and motivational
- Have a sound foundation for software re-engineering and reuse, encompassing both organisational and methodological aspects
- Understand the importance of metrics and be able to apply them in a wide range of software development environments
- Have a basis for the incorporation of formal methods in software development

Prerequisites

The participant is expected to be well-informed in the following areas:

- Software Development Process Models
- Software Requirements and Specification
- Requirements Management
- Analysis Methodologies
- Design Methodologies
- Appropriate texts: Software Engineering (R. Pressman)
  Software Engineering I (Sommerville)

Module Content

I. Software Quality Management and Software Process Improvement

- Analyses of and relationships between ISO 9001:2000, CMMI, ISO 15504 (SPICE) and ISO 12207
- Product Quality (including ISO 9126)
- Software Usability (including ISO 9241)
- Software Configuration Management
- Software Maintenance
  - Characteristics, Maintainability Measurement, Tasks, Costs
- Effect of recent Compliance regulations on the development and maintenance of software
- Implications on Quality and Maintenance of software development trends, e.g. Localisation, Internationalisation, Telecommuting.

II. Localisation

- GILT – Relationships between Globalisation, Internationalisation, Localisation and Translation
- The increasing impact of localisation on Software Engineering and software development projects
- Localisation for a specific locale versus internationalisation

© 2007 Galway-Mayo Institute of Technology

- Localisation skill sets and staffing issues
- The overlaps and anomalies between localisation and the other topics in this module

III. Agile Programming

- Software Development Manifesto
- Agile Methodologies
  - Extreme Programming (XP)/Crystal/SCRUM

IV. Software Re-engineering

- Source code translation
- Program restructuring
- Data re-engineering
- Reverse engineering
- Automating the Re-engineering Process
- Current research in Software Re-engineering

V. Software Reuse

- Motivation for reuse
- Generic reuse development process
- Managing a reuse project
- Measuring the effect of reuse
- Software development with reuse
- Software development for reuse
- Generator-based reuse
- Re-engineering for reuse

VI. Project Management Techniques

- Project team design
- Leadership and motivation
- Group dynamics
- Roles and responsibilities of team members
- Work breakdown, concrete deliverables, milestone setting, milestone reporting, sign-off
- Program Evaluation and Review Technique PERT charts
- Critical Path Method CPM
- Unit development note books, Rate charts

VII. Software Metrics

- Importance and use of software metrics in process improvement and software estimation processes
- Estimation techniques including DELPHI, COCOMO and PUTNAM
- Lines Of Code
- Halstead's metrics
- McCabe's Cyclomatic Complexity
- Function Point Analysis
- Metrics for Object-oriented development
- Metrics for Multi-media applications

© 2007 Galway-Mayo Institute of Technology

VIII. Formal Methods

- Operational Specification, DFD, FSM, PN
- Descriptive Specification, Logical, Algebraic, Z
- Verification, Correctness, Proofs
- Case Studies

Assessment

- Course work and Assignments (30%)
- Written Examination (70%)

Recommended Reading

- Software Quality, a Framework For Success - Sanders & Curran 1994
- Managing the Software Process - W. Humphrey 1989
- Introduction to the Personal Software Process - W. Humphrey 1997
- Introduction to the Team Software Process - W. Humphrey 2000
- Books on Testing by Beizer, Hetzel, Kaner and Myers
- A Discipline for Software Engineering - W. Humphrey 1995
- Software Engineering R. Pressman (5th Edit) 2000
- Software Engineering I. Sommerville (6th Edit) 2000
- Software Re-use - a Holistic Approach - E. Karlsson 1995
- Software Reuse and Re-engineering in Practice - Ed. P. Hall 1992
- The Three R's of Software Automation: Re-engineering, Repository and Re-usability - C. McClure 1992
- Software Inspection - Gilb and Graham 1993
- Software Quality Management and ISO 9000 - M. Jenner 1995
- ISO 9000 for Software Development - C. Schmauch 1995
- The SPIRE Handbook - Centre for Software Engineering 1998
- An Introduction to Formal Specification & Z - Potter, Sinclair & Till 1991

Recommended Journals

- IEEE Transactions on Software Engineering
- Association for Computing Machinery

Module Coordinator

Contact: John Healy (john.healy@gmit.ie)

Last Revision: March 2007.

© 2007 Galway-Mayo Institute of Technology

OBJECT ORIENTED SYSTEMS

MODULE CONTENT…

I. Review of the Object Oriented Paradigm

- Abstraction & Encapsulation
- Generalisation & Polymorphism

II. Classes & Objects

- Class/ Instance Fields & Methods
- Creating & Initialising Objects
- Field Defaults & Initialisers
- Destroying & Finalising Objects
- Object hierarchies and Constructor Chaining
- Inner Classes
- Static & Dynamic Object Introspection. Reflection
- UML notation for classes and objects, UML Datatypes, Multiplicity, Attributes & Operations

III. Composition & Inheritance

- Aggregation, & Association. Uses, context and UML representation, Associations & Navigability, Reifying Associations, N-Ary Associations.
- Inheritance
    - Principle of Substitutability, Subclasses and Subtypes
    - Forms of Inheritance: Specialisation, Specification, Construction, Extension, Limitation, Combination
    - Inheritance in Java: Shadowing Superclass Fields, Overriding Superclass Methods, Modifiers and Inheritance. UML representations.
    - Interfaces & Abstract Classes: Implementing Interfaces, Immutability and Design by Contract, Abstract Classes v Interfaces. Describing Interface and Abstract Classes in UML.
    - Multiple Inheritance: The Deadly Diamond of Death, Using Mixin inheritance to simulate multiple inheritance.
    - Benefits & Costs of Inheritance, Inheritance & Encapsulation, Composition v Ineritance.

IV. Polymorphism

- Static and Dynamic object Binding. Polymorphism & Typing
- Ad Hoc v Universal Polymorphism
- Coercion and Overloading, Parametric Overloading
- Inclusion Polymorphism
- Overriding & Polymorphism, Abstract /Deferred Methods
- Polymorphism, Inheritance & Loose Coupling
- Case Study in Polymorphism: The Collections API
    - The Collection hierarchy and interfaces
    - Set, List, Map implementations.

© 2007 Galway-Mayo Institute of Technology

V. Describing Object Interaction

- Interaction Diagrams: Collaborations, Classifier Roles, Association Roles
- Sequence Diagrams: Messages to Self, Object Creation & Destruction, Guards.

VI. Testing Object Oriented Applications

- Unit Testing and Test Driven Object-Oriented Development, Test Cases and Test Suits, Stubs and Self Shunting, JUnit and the JUnit API.

VII. Class and Package Design Principles

- Class Design
  - Single Responsibility & Open-Closed Principles
  - Liskov Substitution, Dependency Inversion & Interface Segregation
- Package Cohesion & Coupling
  - Reuse-Release Equivalence, Common Closure & Common Reuse Principles
  - Acyclic Dependencies, Stable Dependencies & Stable Abstractions
- Composite Design Patterns: Model-View-Controller, Architectural and Class Level applications of MVC, Producer-Consumer Model.

VIII. Creational Patterns

- Factory & Abstract Factory, Singleton & Concurrency Issues, Builder & Prototype Patterns.

IX. Structural Patterns

- Decorator: Case Study: The Decorator Pattern and the java.io Library
- Façade, Session and Message Façades. The Principle of Least Knowledge.
- Flyweight, Adapters, Bridge, Composite Patterns.
- Proxy: Dynamic, Remote, Smart & Synchronisation Proxies. Copy-on-write, Complexity Hiding.

X. Behavioural Patterns

- Chain of Responsibility, Invocation Handlers.
- Command: Invocation Encapsulation, Marcos, Queuing and Logging.
- Observer: Publisher and Subscribers, Applications of Producer-Consumer Model, Listeners and Push/Pull Models.
- State, Strategy and Template Patterns, The Hollywood Principle. Template v Strategy.
- Visitor: Dynamic Dispatch and Polymorphic Handshakes.
- Interpreter, Iterator, Mediator & Memento Patterns.

XI. Distributed Objects and Strategies

- Overview of architectural approaches. J2EE, CORBA and Web Services. N-Tier Applications and Object-Oriented Development.
- Data Externalisation and Marshalling. Naming Registries & Binding.
- Remote Method Invocation: Remote and Local Interfaces. Object stubs, skeletons. Interface Specification. Serialization & Object Parameters. Passing remote object parameters by reference, Distributed Garbage

© 2007 Galway-Mayo Institute of Technology

Collection, Callback Methods, Dynamic Class Loading, Gateway Objects, JRMP & IIOP.
- Distributed Polymorphism.
- Binding to remote objects using HTTP.
- Stateful and stateless remote objects
- Synchronous and asynchronous distributed object communication. Using message queues.
- Design Patterns for Distributed Architectures: Business Delegate, Business Interface, Home Factory & EJB Command Patterns.

XII. Object Binding with XML
- XML Schemas, Type Scope & Schema Reusability
- JAXB Binding Framework
  - Unmarshalling & Marshalling XML Documents, JAXB Customisation
- Object Persistence with DOM and PDOMs. Using Native XML Databases to persist object state.

XIII. Object-Oriented Databases
- Object-Relational Mapping and Strategies, Case Study: Hibernate.
- Migrating from Relational to Object-Oriented Databases.
- Features of Object-Oriented Databases: Complex Objects, Object Persistence, Transactions, Concurrency
- Object-Oriented DBMS Technologies: Pure OODBMs, Persistent Storage Managers, Object Wrappers.
- Extended RDBMs: Development with Object-Oriented Databases, e.g. O2, Ozone. Object-Oriented DBMS Applications.
- Data Transfer Object (DTO), DTO Factory, Domain Transfer Model, Generic Attribute Access.

Assessment

- Continuous Assessment(s) – 50%
- Final Examination (written) – 50%

Required Reading

- *Object-Oriented Software Engineering* - I. Jacobson 1993
- *The Java Programming Language*, 3rd Ed, Arnold & Gosling, Addison-Wesley, 2000, ISBN: 0201704331
- *Learning UML*, Sinan Si Alhir, O'Reilly, 2003, ISBN: 0-596-00344-7
- *Design Patterns - Elements of Reusuable Object Oriented Software*, Erich Gamma et al, Addison Wesley, 1994, ISBN 0-201-63361-2

© 2007 Galway-Mayo Institute of Technology

Recommended Reading

- *The Object-Oriented Thought Process*, Matt Weisfeld, SAMS, 2000, ISBN: 0672318539
- *Java Programming with CORBA*, Brose, Vogel & Duddy, Wiley, 2001, ISBN: B00005OBU5
- *Java & XML Data Binding*, McLaughlin, O'Reilly, 2002, ISBN:0-596-00278-5
  Booch, G., Jacobsen, I., and Rumbaugh, J. (1997). The UML specification documents. Rational Software Corp., www.rational.com
- Booch, G., Rumbaugh, J. and Jacobson, I. (1999). The Unified Modeling Language user guide. Addison Wesley Longman, Inc. Reading, MA.USA.
- Fowler, M., and Scott, K. (2000): UML Distilled. Addison-Wesley. Reading, MA. USA.
- IEEE Transactions on Software Engineering
- Association for Computing Machinery
- Journal of Object-Oriented Programming

Last Revision: March 2007.

ADVANCED DATABASES

MODULE CONTENT…

I. Relational Databases

- A Review of Relational Theory
- Relational Algebra, Normalisation, SQL Queries
- Query Optimisation
- Function Dependencies
- Inference Axioms (Armstrong & B-Axioms)
- Entity-Relationship Modelling
- Enterprise Data Modelling
- Zachman Charts
- Data Dictionary
- Security and Integrity issues
- Use of CASE tools.
- Design of large database systems.
- Fourth Generation Languages
- Database Middleware
- Gateways to Legacy Databases

II. Database Machines

- Parallel Database Processors.
- Possible Architectures
- Example systems
- Performance
- Data Warehousing

III. Multimedia and Object Orientation

- Concepts and Techniques for Multimedia Data Systems
- Features of Object Oriented and Semantic Databases
- Motivation - OO Concepts Applied to Databases
- Object Oriented Database Systems and Applications

IV Databases and the World Wide Web

- Database Architecture in the Web Environment
- Enterprise DBMS Deployment
- Freeware Data Services

© 2007 Galway-Mayo Institute of Technology

V. Distributed Databases

- Distributed Database Principles
- Advantages and Associated Problems
- Performance Issues, Data placement and Allocation
- Query optimisation
- Concurrency Control and Recovery
- Integrity and Security Issues.

Assessment

- Course work and Assignments (40%)
- Final Examination (60%)

Recommended Reading

- Fundamentals of Database Systems, 5th Edition
  Elmasri & Navathe (2007)

- An Introduction to Database Systems
  C.J Date (2003)

- Database Systems, 3rd Edition
  T Connolly, C Begg & A Strachan (2002)

Recommended Journals

- IEEE Transactions on Software Engineering
- Association for Computing Machinery

Module Coordinator

Contact: Owen Foley (owen.foley@gmit.ie)

Last Revision: March 2007.

NETWORK AND FORENSIC COMPUTING

Objectives

- To provide students with an advanced knowledge of modern data communications and networking technologies. This module will focus in depth on specific areas of current research and development i.e. Scalable High Speed LANs, Internet Protocols, Network Programming and Network Security.
- The course will look at new and emerging technologies and how these are likely to impact on the networks of the future. Advanced areas of networking theory such as protocol design, routing algorithms and multiplayer switched networks will also be covered in some detail.

Technical Prerequisites

Knowledge of C/C++ or Java.

MODULE CONTENT…

I. Internet Core Protocols
- Protocol Design: IP, TCP, UDP, ICMP, BOOTP, DHCP, DNS
- Multicasting Protocols
- Private Network Interconnection (NAT and VPN)
- Classless and Subnet Address Extensions (CIDR)
- Ipv6
- Emerging Standards

II. Multilayer Switching
- Switching V's Routing
- Implementing & Configuring Virtual LANs (VLAN)
- Inter-VLAN Routing
- Enhancing Network Stability, Functionality, Reliability and Performance.

III. Network Programming & Inter-process Communication
- Major Network Services and APIs
- Transport Level Interface Programming
- Advanced Socket Based IPC
- External Data Representation
- Threads
- Input and Output Streams
- Sockets for Clients
- Sockets for Servers
- Multicast Sockets
- Protocol Handlers
- Content Handlers

© 2007 Galway-Mayo Institute of Technology

IV. Network Security

- Cryptographic Theory
- Secret-Key Algorithms
- Public-Key Algorithms
- Digital Signatures
- Authentication and Key Distribution
- Design of Secure Sockets Layer
- Implementing Secure Socket Applications
- Web Security using Secure HTTP
- Emerging Internet Security Protocols

V. Computer Forensic Analysis

- Context of Computer Forensic Analysis
- Legislative Framework
- Collecting Evidence
- Emerging Forensic Analysis Tools & Practices
- Operating Systems and File System Considerations
- The Forensic Workstation
- Root Kits
- Data Hiding Techniques
- Bit-Shifting
- Steganographic Techniques
- Encrypted Files
- Password Recovery

VI. Forensic Analysis of Network Traffic

- Network Intrusions and Attacks
- Direct vs Distributed Attacks
- Buffer Overflows
- Accidental Attacks
- Address Spoofing
- Packet (Header) Spoofing
- ARP Spoofing (Poison ARP)
- DNS Spoofing
- Forensic Tools for Network Investigations (TCPDump, Ethereal and Ethereal)

Assessment

- Course work and Assignments (40%)
- Written Examination (60%)

© 2007 Galway-Mayo Institute of Technology

Recommended Reading

- Internetworking with TCP/IP (Vol. 1 Principles, Protocols & Architecture), 5th Edition. *D. Comer* (2006).
- Computer Networks (4th Edition) , *A. Tanenbaum* (2003).
- Internetworking with TCP/IP (Vol. 2 Design, Implementation & Internals), 5th Edition. *D. Comer* (2006).
- Computer Forensics and Investigations. *B. Nelson, A. Phillips, F. Enfinger, C. Steuart* (2005).
- Building Cisco Multilayer Switched Networks . *R Froom* (2006).
- Building Scalable Cisco Internetworks. *Paquet / Teare* (2006).
- Security in Computing, 4th Edition. *C.P Pfleeger & S.L Pfleeger* (2007)
- Computer Forensics: Principles and Practices. L Volonino, R Anzaldua & J Godwin (2007)

Recommended Journals

- Communications of The ACM
- IEEE Transactions on Communications
- IEEE Network Magazine
- IEEE Communications Magazine
- Computer Communications Review
- Various Internet RFCs (Requests for Comments)

Module Coordinator

Contact: Sean Duignan (sean.duignan@gmit.ie)

Last Revision: March 2007.

MASTERS PROJECT

Part-time Candidates

Part-time participants in the MSc in Computing will be required to complete an individual project as well as four selected subject modules. The project will be substantial and will be of a suitable technical level in line with the overall standards required by HETAC for Masters level degree programmes. It is envisaged that the majority of part-time participants for this degree will work on a project related to their current work, or areas of research relevant to their work environment. This will help foster closer co-operation between the Institute and industry and should make it easier for part-time participants to allocate sufficient time to their project work.

In each case a project supervisor will be assigned from the Institute staff to oversee the project. In some cases, the supervisor may consult with the participant's employer or direct manager to determine how much of the project work can reasonably be done as a part of the participant's normal duties. A flexible approach will be adopted to allow participants access to Institute computing facilities where this need arises.

The project will normally be undertaken after the four required subject modules are completed. In exceptional circumstances it may be commenced after successful completion of a minimum of two modules. The project would normally be completed within one calendar year after its commencement. Extension of the period would be subject to approval of the Course Board. At the enrolment stage a dialogue will be commenced to outline potential project areas.

Project title and definition may be supplied by either the Institute or the candidate themselves. In either case, the project must deal with an original task or problem and must be deemed to be of sufficient technical challenge and depth.

Full-time Candidates

In the early weeks of the programme, each full-time participant will work on the development of an initial research project proposal in a topic or area of interest to them, and aligned to the research focus of their assigned project supervisor. It is envisaged that this proposal, through a series of review meetings, will be appropriately refined and scoped to become a formal project / definition of research study by week 10 of the programme.

Programme of Work

In all cases (full-time and part-time) the definition of the project will include:

- Title
- Project Outline
- Objectives
- Projected time frame of the project
- Resources to be utilised
- Implementation plan
- Relationship (if any) of project to normal work.

© 2007 Galway-Mayo Institute of Technology

It is envisaged that each project will require 450 - 500 hours of effort and must be well documented and verified on completion. The final project report should be comprised of not less than 10,000 words and should be typed and bound for presentation to the course board and external examiners. The report should be of Masters Degree standard and be suitable for publication.

Projects will be assessed by the project supervisor, the Course Board and by external examiners. The following assessment criteria will normally apply:

- Ability to define the objectives of the project
- Demonstration of the participant's understanding of the subject area
- Technical difficulty of project
- Quality of the research of relevant literature
- Contribution to existing subject knowledge
- Innovation and/or potential commercial impact
- Documentation and delivery of project
- Overall structure and impact of project
- The degree to which the objectives have been met.

Most projects will be computer based and should normally involve a substantial amount of software development. This is in keeping with the overall goals of the course to reinforce participant's existing knowledge and to provide relevant and useful new skills in the area of advanced computing techniques and software development.

Recommended Reading

- Successful Dissertations and Theses: A Guide to Graduate Student Research from Proposal to Completion - David Madsen
- Doing your Research Project - J. Bell
- Technical Communications, The Practical Craft - M. Roze
- The Research Project: How to Write it - R. Berry

Last Revision: March 2007.

RESEARCH METHODS FOR COMPUTER SCIENTISTS

Aims and Objectives

This is a short course on research methods specific to research in Computer Science and Informatics. Attendance is compulsory for all full-time research students on the M.Sc in Computing. The module is not formally assessed and does not account for credit accumulation on the programme. It is expected however that this module will facilitate the timely production of a masters project proposal, and will also facilitate applied research and the production of the project dissertation.

This module is designed to provide students with an introduction to the methods used to carry out a postgraduate research project in computing and related disciplines. It is designed for students from a wide variety of backgrounds and aims to help them to develop critical thinking and to learn research techniques. In particular, it will provide them with the skills that they will need to undertake their own research on the MSc programme.

Course Outline

- The Philosophy and Nature of Inquiry.
- The Nature and Objectives of Research.
- Research Design and Planning.
- Research Methodologies for Computing and Informatics.
- Choosing and Planning a Research Project
- Writing a Research Proposal.
- Reviewing the Literature.
- Tools of Research.
- Data Collection and Analysis.
- Managing / Achieving Objectives.
- Writing the Research Report.
- Presenting Data, Drawing Conclusions, Identifying Future Work.
- Presenting Research Findings.

Module Coordinator

Contact: Sean Duignan (sean.duignan@gmit.ie)

Last Revision: March 2007.

Elective Module Information & Syllabus Content

BIOINFORMATICS

MODULE CONTENT…

I. Molecular Biology Primer

- Nucleotides, orientation, base-pairing, central dogma, promoter sequences, genetic code, open reading frames, introns and exons.
- Restriction enzymes, PCR, cloning and probing.

II. Databases and Data Formats

- Overview of main biological databases, FASTA, NCBI sequence identifier syntax, Non-redundant database syntax.
- Genbank flat file format, field definitions and feature tables, Swiss-Prot flat file format and field definitions. XML formats.
- The BioJava API.

III. Pairwise Sequence Alignment & Sequence Similarity

- Global and local sequence alignments alignments, Needleman-Wunch and Smith-Waterman algorithms, dynamic programming, multiple sequence alignment.
- Information theory and Shannon's entropy, amino acid similarity, scoring matrices, target Frequencies – Lambda and H, Karlin-Altschul equation and BLAST statistics.
- The BLAST algorithm, seeding, extension and evaluation, BLAST reports and statistics analysis.

IV. Phylogenetics

- Phylogenetic trees, distance matrix methods, neighbour's relation and joining methods, character-based methods for phylogenetics, parsimony, tree confidence, molecular phylogenies.

V. Genomics and Gene Recognition

- Prokaryotic genomes and gene structure, promoter elements, open reading frames, conceptual translation, termination sequences, eukaryotic gene structure, promoter elements, protein binding sites.
- Hidden Markov Models, decoding algorithm and parameter estimation, profile HMM alignment.

VI. Genome Assembly

- DNA sequencing strategies, sequence reconstruction, fragment assembly and graph theory.

© 2007 Galway-Mayo Institute of Technology

Assessment

- Continuous Assessment(s): 40%·
- Final Examination (written): 60% - 6 Questions (Answer 4 in 3 Hours)

Pass Criteria and Compilation of Final Mark or Grade

- 40% Overall
- 30% or more of available marks for each component.

Required Reading

- Fundamental Concepts of Bioinformatics, Krane & Raymer, 2003, Pearson Education, ISBN: 0-321-10922-X
- Discovering Genomics, Proteomics and Bioinformatics, Campbell and Heyer, Cold Spring Harbour Laboratory Press, 2003. ISBN: 0-08053-4722-4.
- An Introduction to Bioinformatics Algorithms, Jones & Pevzner, 2004, MIT Press, 0262101068

Recommended Reading

- Algorithms on Strings, Trees and Sequences – Computer Science & Computational Biology, Gusfield, Cambridge University Press, 1997, ISBN: 0-521-68519-8
- Biological Sequence Analysis – Probabilistic Models of Proteins and Nucleic Acids, Durbin et al, Cambridge University Press, 2002, ISBN: 0-521-62041-3
- Journal of Bioinformatics
- Journal of Computational Biology

Module Coordinator

Contact: John Healy (john.healy@gmit.ie)

Last Revision: March 2007.

© 2007 Galway-Mayo Institute of Technology