TABLE OF CONTENTS

I. INTRODUCTION

The Bioinformatics program of Future University Sudan combines the academic courses and their applications. The program prepares the students with the necessary academic and professional experiences that enable them to develop and practice various computer principles and practices. The existing program also incorporates emphasis on related Topics to the Bioinformatics field and the environment.

Bioinformatics focuses upon those aspects of the computer field that have developed a well-defined set of abstract concepts and principles. Thus, the objective of a Master in Bioinformatics is to demonstrate a strong comprehension of those abstract concepts and principles and the theory of the field and to implement, demonstrate and test this theory via a computer. The Bioinformatics Department of Future University Sudan encourages students to tailor their upper-level further studies to master degree program.

The goal of the Bioinformatics with concentration of Future University Sudan is to produce graduates whose strong, balanced and general preparation in Bioinformatics prepares them for positions in the workplace or for graduate study. Graduates of the Bioinformatics with master degree will have the analytical, experimental and professional skills needed to identify, formulate and solve scientific and technical problems throughout their careers; able to address technical, societal and ethical dimensions of computing; and have an awareness of the importance of professional and personal integrity, cultural awareness and ethical behavior in their Careers.

II. PHLOSOPHY OF THE PROGRAM

Master of Science in Bioinformatics Degree Program of Future University provides the students with an opportunity to enhance their knowledge, skills, and experience to become successful practitioners and leaders in the field of Bioinformatics.

The Program is designed to prepare the students to pursue their careers as Bioinformatics professionals. It is suitable for students with a background in computing and/or biology. The Program is organized to enhance students' understanding of the theories, concepts and practices of Bioinformatics, enabling them to develop new skills and competencies.

III. AIMS OF THE PROGRAM

The aims of the program are to provide opportunities for suitably qualified graduate students to acquire skills and expertise necessary to undertake research and development in the field of Bioinformatics. The courses in the M.Sc. Bioinformatics enable students to acquire expertise, and enhance their technical skills in these fields. This program prepares the graduate student to successfully handle problems requiring in-depth knowledge principles and processes in these fields. Students who successfully complete this Program will be qualified to assume responsible positions in industry and government at the research, planning and development levels. This program is planned to achieve the following goals:

- To develop an understanding of the theoretical concepts and principles, underlying the science of bioinformatics.
- Application of the theories to the practices of genomic computing. Assume responsible positions and apply skills in industry and government at the research, planning, and development levels,

- To develop a strong foundation and enthusiasm in students for the growth of their skills and knowledge in Bioinformatics through independent research and study.

IV. LEARNING OUTCOMES

Upon the completion of the program, the student is expected to acquire intellectual and transferrable skills. Following are the learning outcomes for M.Sc. Bioinformatics program:

M.Sc. Bioinformatics

The emphasis of the program is on both theoretical and practical techniques for the design and development of bioinformatics applications, enabling graduates to apply their knowledge and skills in a variety of bioinformatics software development processes.

V. CAREER OPPORTUNITIES

Bioinformatics is widely considered to be one of the most dynamic fields in terms of market growth, and accordingly, job prospects for Computer graduates are excellent. A master degree in Bioinformatics is held in high regard in industry and is an important asset for launching a successful international career.

As a graduate in Master of Science in Bioinformatics he/she is qualified to compete in the following:

- Bioinformatics System Analyst
- Bioinformatics Analyst/Programmer
- Bioinformatics Analyst/Scientist

- Bioinformatics Research Specialist
- Bioinformatics System Design Analyst/Programmer
- Bioinformatics Research Analyst
- Bioinformatics Information System Developer
- Bioinformatics Service Analyst/Programmer
- Bioinformatics Quality Assurance Supervisor
- Bioinformatics Analyst/Programmer
- Bioinformatics Information System Administrator
- Biotechnologist
- Academician

VI. ENTRY REQUIREMENTS

Admission Requirements for Master Program in Bioinformatics

Bachelor Degree Honors (first or second) in Bioinformatics or another equivalent.

Obtain a bachelor's degree in Bioinformatics (a very good grade) from the Faculty of Bioinformatics and Information Technology or any college and experience with a similar field.

Access to Higher Diploma in Bioinformatics (GPA) 3.0 minimum from the Faculty of Bioinformatics and Information Technology or any similar.

Obtain a bachelor degree in a related area of Bioinformatics Cognitive such s: Information Technology, Information Systems, Computer Engineering and Mathematics.

The applicant for the program should present an authenticated Official Transcript of Records (OTR) from the institution granting him/her the bachelor degree.

VII. GENERAL REGULATIONS

Introduction:

1. Master's degrees offered by Course work only.

Registration:

2. Candidates must register as full-time students before proceeding to the Master's degree.

3. The normal qualification for proceeding to the Master's Degree shall be Honors degree from Faculty of Computer Science of The Future University or its equivalent. A candidate who holds another qualification from this University or another University may be required to pass a qualifying examination.

Procedure:

4. Each applicant for registration for the Master Degree must complete the prescribed application form and submit it to the Registrar of the Faculty of Post Graduate Studies.

5. Registration will lapse if the student does not attend his course of study with ONE month. The student is required to notify the Registrar of the Faculty of Post Graduate Studies in writing that he/she has commenced his/her study within the period specified above.

Course of Study:

6. No student will be allowed to continue with his registration after the lapse of six calendar years. In special cases the Board of the Faculty of Post Graduate Studies may approve the extension of this period.

7. The course of study must be pursued continuously, except by special permission of the Board of the Faculty of Post Graduate Studies.

8. A student who is employed must inform the University of the nature of his employment at the time of application. Any change of occupation during a course must be notified to the Dean of the Faculty of Post Graduate Studies.

Dissertation:

9. The title of dissertation must be submitted within THREE months after finishing the course work of EIGHTEEN months from the date of registration. The title of the dissertation may be changed. However the candidate will not be permitted to change the title of dissertation not later than SIX months before the date of submission of the dissertation.

10. The student, on completing his course of study, must submit a dissertation which complies with the following conditions:

    The dissertation must be written in English. In special cases, the use of other languages, if deemed necessary, may be approved by the Faculty of Post Graduate Studies Board at the time of registration.

    The greater portion of the work submitted therein must have been done subsequent to the registration of the student as a candidate for the Master degree.

    It must form a distinct contribution to knowledge of the course and afford evidence of independent critical and/or analytical power.

    The literary presentation must be satisfactory, and suitable for publication, either as submitted or in an abridged form.

11. A student should not submit a dissertation for which a degree has been conferred on him from this or any other University, but he shall not be precluded from incorporating work which he has already submitted for a degree in a dissertation covering a wider field provided that he shall state the extent of such work in his dissertation.

12. The candidate should prepare eighth copies of an abstract of the dissertation in both Arabic and English of not more than one thousand words.

13. Every candidate will be required to present to the Registrar of the Faculty of Post Graduate Studies in printed matter (8) EIGHT pound copies of the dissertation and (8) EIGHT copies of the abstract. The Registrar shall

send one copy to each examiner and retain the additional copies for further distribution.

The examiners are required to return such copies with their reports.

On successful award, the Registrar of the Faculty of Post Graduate Studies should send:

1. Two copies of the University Library.
2. One copy to Faculty of Post Graduate Studies Library.
3. One copy for the employer (if any).
4. One copy for the National Training Centre, where applicable.
5. One copy to the Central Documentation Centre.
6. One copy to the department.
7. One copy to the supervisor.

The registrar shall retain one copy of any thesis which is rejected or referred and return the rest to the candidate.

14. If a successful candidate publishes his dissertation or a part thereof, the fact of approval of the Master Degree of The Future University must be mentioned in a appropriate place.

Examination:

(To be read with the Post-Graduate Degrees examination regulations).

Examiners:

15. All examiners will be appointed by the Board of the Faculty of Post Graduate Studies, on recommendation of the Departmental and Faculty Research Board concerned. The Faculty of Post Graduate Studies shall appoint three examiners at least one of whom shall be an external examiner and one an internal examiner. The external examiner shall be a recognized authority in the field of study. When a candidate has had two supervisors,

only in exceptional cases may both supervisors act as his internal examiners.

16. Each examiner shall read the dissertation and submit a separate strictly confidential report to the Faculty of Post Graduate Studies which must include:

a. Recommendation, (1) Accepted, (2) Referred, or (3) rejected.

b. Ground for recommendation.

c. Comprehensive report on the dissertation

Giving evaluation of the following:

i. Review of the literature.

ii. Methods.

iii. Results obtained.

iv. Analysis, interpretation and discussion of results.

v. Any other comments.

vi. Conclusion drawn.

17. The oral examination should be conducted only after receipt of all examiners reports by the Faculty of Post Graduate Studies. Such reports shall be made available to the Board of Examiners.

18. The Board of Examiners who conduct the oral, shall send a joint report to the Faculty of Post Graduate Studies stating:

i. Whether it is recommended that the candidate should pass, fail or be referred for a period not exceeding NINE months after which the candidate may resubmit his dissertation for examination.

ii. The grounds of their recommendation.

iii. The candidate's contributions (if any), to the advancement of his subject.

References in Written Examinations:

19. In Master Degrees where a written examination is specified students are allowed to:

a. Take supplementary if they fail in not more than one third of their subjects.

b. Resit the whole examination if he fails in more than one third and up to half of his subjects.

Any student who fail in more than half of his subjects or his supplementary or resit will not be allowed to continue with his studies.

Award of the Degree:

20. The Faculty of Post Graduate Studies Board will consider all the relevant reports and other items and:

i. Recommend the names of successful candidates to the Senate for award.

ii. Decide cases of supplementaries and resit (if any) and cases of referral.

iii. Decide cases of rejection and failures.

VIII. PROGRAM STRUCTURE

Summary of Requirements:

The Master of Science in Bioinformatics is both a research oriented requiring a thesis for successful completion or yields a professional degree with significant emphasis on a project and practical experience in industry. The later expects to produce a practitioner who can rapidly assume a position of substantial responsibility. The Future University proposed curriculum is intended to meet the increasing need of highly skilled scientists, systems analysts and software developers. It is a professional degree program and will not prevent qualified graduates to continuing their studies towards a doctoral degree.

The Master of Science in Bioinformatics is composed of **three semesters of course work and one semester of capstone course** (project) for a total of **39 credit** hours.

In totality, the Master of Science in Bioinformatics has 33 credit units of coursework and 6 units of Project Development:

Core Courses	15 credit units
Specialization Courses	18 credit units
Master's Project	6 credit units

Semester 1

Courses	Credit Hours
Core I	3
Core II	3
Core III	3
Specialization I	3
TOTAL	**12**

Semester 2

Courses	Credit Hours
Core IV	3
Core V	3
Specialization II	3
Specialization III	3
TOTAL	**12**

Semester 3

Courses	Credit Hours
Specialization IV	3
Specialization V	3
Specialization VI	3
TOTAL	**9**

Semester 4

Course	Credit Hours
Project	6
TOTAL	**6**

IX. DISTRIBUTION OF THE COURSES

Core Courses:

These courses are required for the students enrolled in MSc Bioinformatics program. These are common to the three tracks.

	TITLE	CREDIT HOURS
MSCS01	Theory of Computing	3
MSCS02	Advanced Computer Architecture	3
MSCS03	Advanced Algorithm Design and Analysis	3
MSCS04	Advanced Operating Systems	3
MSCS05	Principles of Programming Languages	3

Bioinformatics Specialization Courses

COURSE CODE	TITLE	CREDIT HOURS
MSCBI01	Genomics & Gene Expression	3
MSCBI02	Proteomic Informatics	3
MSCBI03	Simulating Biological Systems	3
MSCBI04	Informatics for Metabolomics	3
MSCBI05	Data Analysis & Essential Statistics	3
MSCBI06	Data Integration and Interaction Networks	3

Master's Project:

Students in the Master of Science in Bioinformatics (MSCBio) program are required to do a Master's Research Project. To opt for a Master's Research Project, a student must have at least three B Grades in all core courses. Students who have not completed their core course requirements or low grades, they will be not allowed to register for the research project or required to repeat the courses with low grades in order to satisfy these requirements. To complete a research project student must have to follow these steps:

Step1 - Find an Advisor

The student must find a faculty member who is willing to work with the student and supervise the research project. The advisor and the student should formulate a plan for the work to be done. A Master's Research Project can consist of a high-quality paper or piece of software.

Faculty members will customarily agree to serve as a research project advisor only when they have an interest in the topic and a high degree of confidence in the ability of the student to complete the thesis or project. Students cannot do a thesis or project unless they find a faculty advisor.

Step 2 - Form a Committee

Once the work to be done has been planned, a defense committee must be formed. Once the members of the committee have been identified, the student must prepare a written description of the research project.

Step 3- Carry out the Work and Defend It

Once the research project work has been completed and a document describing the results prepared, the student must schedule an oral defense of this work.

	TITLE	CREDIT
MSCP-07	Master's Project	6

DISTRIBUTION OF COURSES

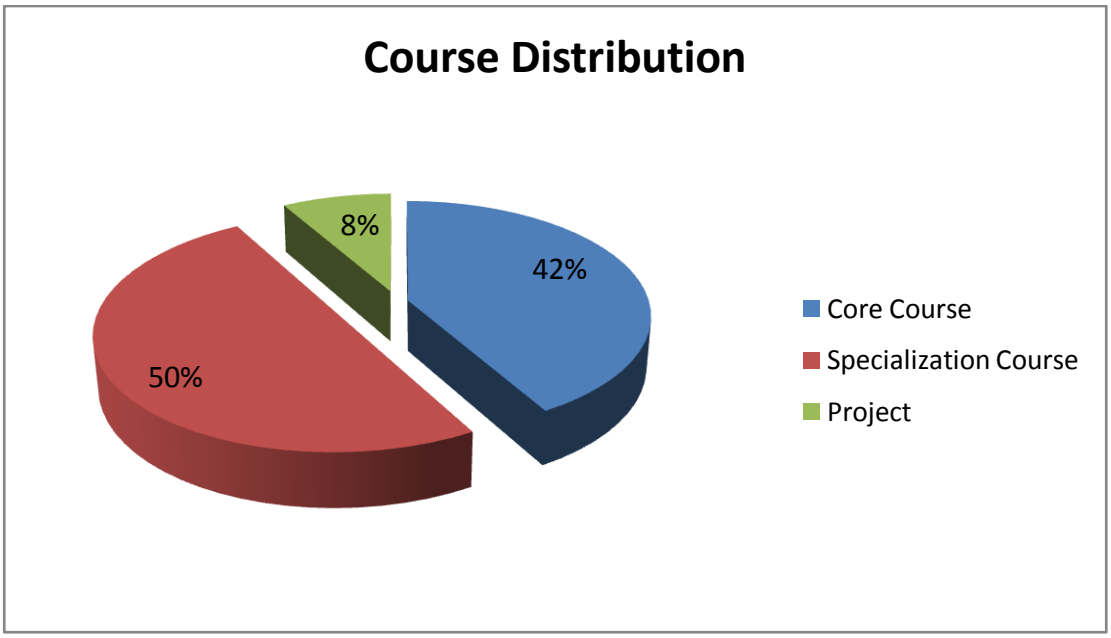

Figure 1: Percent Distribution of Courses

X. COURSE DESCRIPTION

Core Courses

Course Code	MSCS01
Course Title	Theory of Computing
Credit Hours	3
Prerequisites	None

Aims	To provide the student with fundamental knowledge about complexity classes and the methodology of complexity theory.
Learning Outcomes	By the end of the course, students should be able to: • Identify and describe the properties and differences between various computational models, including the Turing machine and the random access model. • Apply the Master theorem to obtain asymptotic terms for recurrence relations. • Identify and characterize the time and space complexity hierarchies. They should be able to apply both the hierarchy theorem and the reachability method. • Classify a given problem into its complexity class using methods such as reduction and polynomial transformations. • Identify the characteristics of example problems classified as NP Complete. They should be able to, for a given similar problem, prove that it is NP Complete.
Course outlines	**Computational Models and the Analysis of Algorithms** The Turing Machine Model, The Random Access Model of Computation, Comparison of different Models, Basic Analysis of Algorithms, O-notation, Growth of a Function. **Advanced Analysis of Algorithms** Recurrence Equations, Analysis of Recursive Algorithms, Master Theorem, A General Form of Recursive Equations.

	Fundamental Complexity Classes Time and Space Complexity, The Complexity Hierarchy, Linear SpeedUp, Space Bounds, The Hierarchy Theorem, The Reachability Method. **Classification of Problems** Reduction, Completeness, Examples for polynomial transformations (SAT, 3-SAT, Hamilton Circuit, Partition), Cooks Theorem **P, NP, NPC** Example Problems from NPC (Satisfiability, Graph-theoretic Problems, Set Problems), Proving NP-completeness, Proving Strong NP-completeness, Coping with NPC (Approximation, Randomization), The Polynomial Hierarchy, On P vs NP **Introduction to Advanced Complexity Theory** Inside P (LOGSPACE, Parallel Computation), Beyond NP (PSPACE, EXPTIME), #P, Circuit complexity, Information (Kolmogorow) Complexity
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%

References	
Text Books	Michael S. Sipser, Introduction to the Theory of Computation, 2nd Edition, Thomson Course Technology, 2006.
References	Michael R. Garey, David S. Johnson, Computers and Intractability, W. H. Freemann and Company, 1999. Michael Sipser, Introduction to the Theory of Computation, PWS Publishing Company, 1997.

Course Code	MSCS02
Course Title	Advanced Computer Architecture
Credit Hours	3
Prerequisites	None
Aims	This course provides a thorough discussion of the fundamentals of computer organization and architecture and relates this to contemporary design issues.
Learning Outcomes	By the end of the course, students should be able to: • Understand the details of computer architecture and organization. • Explain about the primary computer components such as CPU, memory and input output devices. • Identify and explain the purpose of different levels of memory.

	• Explain different instructions for a processors
Course outlines	**Architecture And Machines** Some definition and terms, interpretation and microprogramming. The instruction set, Basic data types, Instructions, Addressing and Memory. Virtual to real mapping. Basic Instruction Timing. **Time, Area And Instruction Sets** Time, cost-area, technology state of the Art, The Economics of a processor project: A study, Instruction sets, Professor Evaluation Matrix **Cache Memory Notion** Basic Notion, Cache Organization, Cache Data, adjusting the data for cache organization, write policies, strategies for line replacement at miss time, Cache Environment, other types of Cache. Split I and D-Caches, on chip caches, Two level Caches, write assembly Cache, Cache references per instruction, technology dependent Cache considerations, virtual to real translation, overlapping the Tcycle in V-R Translation, studies. Design summary. **Memory System Design** The physical memory, models of simple processor memory interaction, processor memory modeling using queuing theory, open, closed and mixed-queue models, waiting time, performance, and buffer size, review and selection of queueing models, processors with cache.

	Concurrent Processors Vector Processors, Vector Memory, Multiple Issue Machines, Comparing vector and Multiple Issue processors. **Shared Memory Multiprocessors** Basic issues, partitioning, synchronization and coherency, Type of shared Memory multiprocessors, Memory Coherence in shared Memory Multiprocessors.
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%
References	
Text Books	Advance computer architecture by Hwang & Briggs, 1993.
References	Pipelined and Parallel processor design by Michael J. Fiynn – 1995, Narosa.

Course Code	MSCS03
Course Title	Advanced Algorithm Design and Analysis

Credit Hours	3
Prerequisites	None
Aims	The course aims to study the time and space complexity of computer algorithms. The student will learn different paradigms for solving. This together with the repertoire of algorithms presented in the course will provide the student with the ability to design and develop new algorithms. This information can be used to decide if a given algorithm can be adopted in particular situations where constraints are placed on the problem.
Learning Outcomes	By the end of the course, students should be able to: • Develop efficient computer algorithms in terms of space and time complexities. • Choose among different paradigms the best one for problem solving • Appreciate the complexity of non-polynomial time algorithms and also acquire advanced programming skills • Compare algorithms with respect to time and space requirements.
Course outlines	**Brief Review** Graphs, Sets and disjoint sets, union, sorting and searching algorithms and their analysis in terms of space and time complexity. **Divide and Conquer** General method, binary search, merge sort, qick sort, selection sort, Strassen's matrix multiplication

	algorithms and analysis of algorithms for these problems. **Greedy Method** General method, knapsack problem, job sequencing with dead lines, minimum spanning trees, single souce paths and analysis of these problems. **Dynamic Programming** General method, optimal binary search trees, O/I knapsack, the traveling salesperson problem. **Back Tracking** General method, 8 queen's problem, graph colouring, Hamiltonian cycles, analysis of these problems. **Branch and Bound** Method, O/I knapsack and traveling salesperson problem, efficiency considerations. Techniques for algebraic problems, some lower bounds on parallel computations. **NP Hard and NP Complete Problems** Basic concepts, Cook's theorem, NP hard graph and NP scheduling problems some simplified NP hard problems.
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20%

	• Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%
References	
Text Books	• Fundamental of Computer algorithms, Ellis Horowitz and Sartaj Sahni, 1978, Galgotia Publ., • Introduction To Algorithms, Thomas H Cormen, Charles E Leiserson And Ronald L Rivest: 1990, TMH
References	• The Design and Analysis of Computer Algorithm, Aho A.V. Hopcroft J.E., 1974, Addison Wesley. • Algorithms-The Construction, Proof and Analysis of Programs, Berlion, P.Bizard, P., 1986. Johan Wiley & Sons, • Writing Efficient Programs, Bentley, J.L., PHI • Introduction to Design and Analysis of Algorithm, Goodman, S.E. & Hedetnieni, 1997, MGH. • Introduction to Computers Science- An algorithms approach , Jean Paul Trembley, Richard B.Bunt, 2002, T.M.H. • Fundamentals of Algorithms: The Art of Computer Programming Voll, Knuth, D.E.: 1985, Naresh Publ.

Course Code	MSCS04
Course Title	Advanced Operating Systems
Credit Hours	3
Prerequisites	None
Aims	To build sound foundation, the design and

	implementation of a typical operating system
Learning Outcomes	By the end of the course, students should be able to: • Understand different types of operating system environments • Understand issues involved in design and implementation of a typical operating system
Course outlines	**SECONDARY-STORAGE STRUCTURE** Disk Structure, Disk Scheduling, Disk Management, Swap-Space Management, Disk reliability. **NETWORKING AND DISTRIBUTED PROCESSING** Synchronisation, Naming user, system and physical layers. **DISTRIBUTED PROCESS MANAGEMENT** Remote operations on processes and migration of processes, message passing, remote procedure calls and communication for transaction. **RESOURCES MANAGEMENT** Structures of resource management systems, load sharing, load balancing and load balancing algorithms, resource protection. **SECURITY** Communication security, data encryption, key management, User authentication, Protection Mechanisms as threads, real-time systems, multiprocessor scheduling, process migrations and

	security. **CASE STUDIES** UNIX, Window NT-History, Design principles, Programmer interface, User Interface, Process Management, Memory Management, File System, I/O System, Interprocess Communication.
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%
References	
Text Books	Abraham Silberschatz, Operating System Concepts, 6th Edition. John Wiley, 2002.
References	• Randy Chow and Theodore Johnson, Distributed Operating Systems and Algorithms, Addison Wesley, 1997 • William Stallings Operating Systems, 2/e, Prentice Hall Engineering, Science & Math, 1995. • Andrew S. Tanenbaum, Operating System: Design and Implementation, 2/e, Prentice Hall, 1996 • Goscinki, A., Distributed Operating Systems: The logical Design, Addison Wesley, 1997

Course Code	MSCS05
Course Title	Principles of Programming Languages
Credit Hours	3
Prerequisites	None
Aims	To provide students with a good grasp of the structure of programming languages, which will improve their understanding of different language implementations and also their capability to master new languages effectively
Learning Outcomes	By the end of the course, students should be able to: • Identify the concepts of structure of programming languages correctly. • Demonstrate different language implementations correctly and effectively. • Improve their understanding of various programming languages' paradigms and implementations
Course outlines	**Introduction** Syntactic and semantic rules of a Programming language, Characteristics of a good programming language, Programming language translators compiler & interpreters , Elementary data types – data objects, variable & constants, data types, Specification & implementation of elementary data types, Declarations ,type checking & type conversions , Assignment & initialization, Numeric data types,

	enumerations, Booleans & characters. **Structured data objects** Structured data objects & data types , specification & implementation of structured data types, Declaration & type checking of data structure ,vector & arrays, records Character strings, variable size data structures , Union, pointer & programmer defined data objects, sets, files. **Subprograms and Programmer Defined Data Types** Evolution of data type concept, abstraction, encapsulation & information hiding, Subprograms, type definitions, abstract data types. **Sequence Control** Implicit & explicit sequence control, sequence control within expressions, sequence control within statement, Subprogram sequence control: simple call return, recursive subprograms, Exception & exception handlers, co routines, sequence control. **Data Control** Names & referencing environment, static & dynamic scope, block structure, Local data & local referencing environment, Shared data: dynamic & static scope. Parameter & parameter transmission schemes. **Storage Management** Major run time elements requiring storage ,programmer and system controlled storage

	management & phases , Static storage management , Stack based storage management, Heap storage management ,variable & fixed size elements. **Programming Languages** Introduction to procedural, non-procedural, structured, functional and object oriented programming language, Comparison of C & C++ programming languages.
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%
References	
Text Books	• Programming languages Design & implementation by T.W. .Pratt, 1996, Prentice Hall • Programming Languages – Principles and Paradigms by Allen Tucker & Robert Noonan, 2002,
References	• Fundamentals of Programming languages by Ellis Horowitz, 1984, Galgotia publications (Springer Verlag) • Programming languages concepts by C. Ghezzi, 1989, Wiley Publications. • Programming Languages – Principles and Pradigms Allen Tucker , Robert Noonan 2002

Bioinformatics Specialized Courses

Course Code	MSCBIO1
Course Title	Genomics & Gene Expression
Credit Hours	3
Prerequisites	None
Aims	To introduce the techniques that have given rise to the genomic data now available, and develop skills and understanding in the bioinformatics approaches that facilitate evaluation and application of these data.
Learning Outcomes	By the end of the course, students should be able to: • Explain the mode of operation of the most common analytical techniques used in the acquisition of genomic sequence and expression data • Appreciate the challenges of dealing with sequence data and be able to identify and apply appropriate software tools to tackle these challenges • Demonstrate critical awareness of current practices and recognise the relative strengths and weaknesses of the techniques covered and how these relate to the quality of the biological findings that result • Utilise a number of sequence databases and related software tools for sequence analysis, and interpret the output from these tools • Utilise a number of gene expression databases and analyse datasets • Understand the basic concepts and limitations of major NGS technologies.

Course outlines	• Introduction to contemporary sequencing techniques • Sequence databases and gene expression databases (GEO, ArrayExpress) • Sequence data analysis platforms (e.g. Galaxy) • Sequence assembly and quality control • Transcriptome informatics • Ethical considerations in genomics • Next Generation Sequencing (NGS) Technology.
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%
References	
Text Books	• •
References	• • •

Course Code	MSCBIO2
Course Title	Proteomic Informatics

Credit Hours	3
Prerequisites	None
Aims	To provide the students with an awareness of the current trends in proteomics and the crucial role that bioinformatics plays within this field.
Learning Outcomes	By the end of the course, students should be able to: • Explain the mode of operation of the most common analytical techniques used in the acquisition of proteomic data • Demonstrate critical awareness of current practices and recognise the relative strengths and weaknesses of the techniques covered and how these relate to the quality of the data acquired • Discover information using bioinformatics tools and effectively apply the information to biological problems.
Course outlines	• Introduction to practical proteomics (qualitative & quantitative) • Proteomics repositories (PRIDE, PeptidAtlas, etc.) • Protein/peptide identification algorithms (Mascot, X!Tandem, OMMSA) • Tools for quantitative proteomic data (iTRAQ, SILAC, SRM, etc).
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20%

	Midterm Exam: 20% Final Exam: 40%
References	
Text Books	• •
References	• • •

Course Code	MSCBI03
Course Title	Simulating Biological Systems
Credit Hours	3
Prerequisites	None
Aims	To develop the concepts of integrated data towards simulation of systems and to introduce the potential applications of robust simulation.
Learning Outcomes	By the end of the course, students should be able to: • Utilise systems modelling software for the simulation of systems and system interactions • Describe the potential of systems modelling to impact on commercial interests • Outline the role of systems biology towards the 3Rs • Discuss the limitations of existing approaches and the necessary directions research must take to realise the promise.
Course	• Networks dynamics (from classic approaches e.g.

outlines	ODE and stochastic algorithms to rule based systems) • SBML • Modelling systems ie. eCell • The concept of 3Rs and systems modeling • In silico approaches to toxicity and efficacy testing.
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%
References	
Text Books	• •
References	• • •

Course Code	MSCBI04
Course Title	Informatics for Metabolomics
Credit Hours	3
Prerequisites	None
Aims	To explore the analytical and statistical techniques that are central to the field of metabolomics, and introduce the emerging technologies that will generate

	yet more data in the future.
Learning Outcomes	By the end of the course, students should be able to: • Explain the mode of operation of the most common analytical techniques used in the acquisition of metabolomic data • Explain and carry out the mathematical procedures covered during the course, to derive biological relevant information from metabolomics data set • Demonstrate critical awareness of current practices and recognise the relative strengths and weaknesses of the techniques covered and how these relate to the quality biological findings that result.
Course outlines	• Introduction to Metabolomics • Introduction to NMR, LC-MS and GC-MS • An introduction to R. • Multivariate classification (PLS-DA, SVMs, ANNs) • Multiway analysis (PARAFAC) • Compound identification (e.g. spectral library searching) and biomarker discovery.
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%

References	
Text Books	• •
References	• • •

Course Code	MSCBI05
Course Title	Data Analysis & Essential Statistics
Credit Hours	3
Prerequisites	None
Aims	To provide an overview of important concepts in statistics and exploratory data analysis. The course introduces the main concepts in analysing biological datasets using the R environment, as well developing bespoke scripts for multivariate analysis such as principal component analysis and hierarchical clustering.
Learning Outcomes	By the end of the course, students should be able to: • Devise advanced R programs to meet given specifications • Discuss the applicability of different statistical techniques and be able to implement them appropriately • Demonstrate a thorough understanding of the need to integrate statistics into experimental protocol design.
Course	• Introductory statistics – averages, variance and

outlines	significance testing • Exploratory data analysis (PCA, HCA) • Advanced topics in R
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%
References	
Text Books	• •
References	• • •

Course Code	MSCBI06
Course Title	Data Integration and Interaction Networks
Credit Hours	3
Prerequisites	None
Aims	To introduce systems biology, systems methodologies, the most important bioinformatics software tools, and explain the algorithms that underpin them.

Learning Outcomes	By the end of the course, students should be able to: • Utilise systems software for the visualisation of systems and system interactions • Identify the main challenges facing systems biology • Use available resources for data integration • Design, normalise and implement databases for experimental datasets • Design simple database front interfaces using Java EE • Identify main data standards protocols for systems biology • Discover systems relationships between data using bioinformatics tools and approaches.
Course outlines	• Introduction to Systems Biology • Systems Biology data standard protocols • Database design and normalization • Designing database front end using Java EE • Introduction to interaction networks • Data Integration using Taverna • Data visualisation using Cytoscape.
Teaching Methods	28 Hours of Lecture 28 Hours of Tutorial
Assessment Methods	Coursework: 40% • Assignments/Quizzes: 20% • Project/Case Study : 20% Midterm Exam: 20% Final Exam: 40%

References	
Text Books	• •
References	• • •

XI. TEACHING STRTEGIES AND TECHNIQUES

Helping students learn is a big challenge for the instructors/teachers in the faculty of Computer Science. Identifying effective teaching strategies is a task that lies ahead of the Bioinformatics teacher. These strategies help the students understand and implement Bioinformatics principles and go beyond the knowledge level to higher levels of thinking to be able to offer solutions to problems that require engineering knowledge.

1. PRACTICAL EXAMPLES: Connecting Theory with Applications

Practical examples can be included at all levels of the Bioinformatics curriculum. When determining examples to be used for instruction it is important to make the examples as clear and straight forward as possible. The key is to make the examples as simple as possible, and to make sure that they isolate the desired principle.

2. SHOW AND TELL: Reversing Student Roles

Putting the students into the role of a teacher makes the students look deeper into the assigned problem. The students will be forced to clarify their thinking and understanding since they must explain to their peers the phenomena that they are demonstrating. In searching for examples outside of

class or for demonstrations that can be performed in class, the students will be compelled to look for connections between theory and practical application.

3. *CASE STUDIES: Bringing "Real-Life" Scenarios into the Classroom*

Students are usually given written material regarding a case and asked to read it and answer a series of questions pertaining to various aspects of the case. The students can be required to work either individually or in groups.

4. *GUIDED DESIGN PROJECTS: Introducing Practical Design Experience in Classrooms*

The following are some tips to remember when using a guided design project:

a) Realize that the project is not as important as the thought processes that go into determining a design. It is not important that the students determine an optimum design. What is important, however, is that they experience the design process.
b) Starting before the semester, determine the scope of the projects and the goals for the class. It is important that the scope of the project is reasonable, and care must be taken to ensure that the students are not overloaded.
c) When possible, divide the design into sections. This spreads the work for the students and the grading duties over the semester. Design teams of two or three students are frequently used. This allows for in-depth projects, reduces the grading load of the TA, and promotes interaction among the students.

5. *OPEN-ENDED LABS: Making Students Think Deeper*

Laboratory classes can be made open-ended to varying degrees depending on a number of factors including the autonomy that the instructor has in changing the course structure, the facilities available, and the degree of difficulty that the instructor perceives to be appropriate for the class.

6. THE FLOWCHART TECHNIQUE: Organizing the Flow of Thought

The technique of flowcharting, as applied to a classroom scenario, is a tool for precisely and concisely representing the flow of information among various stages in the development of a theoretical concept or in the formulation or analysis of an engineering problem. Flowcharts are a tool to organize the flow of logic and thought in a classroom, much in the way that flowcharts help in presenting the flow of materials between various units of an industrial process.

7. OPEN-ENDED QUIZZES: Moving Students Away From Memorization

The open-ended quiz is intended to stimulate students' creativity and to help students to think deeply about the material covered in lectures. In contrast, straightforward “Given this, calculate that” or “Plug and chug” type of quizzes merely encourage students to memorize equations and formulas.

8. BRAINSTORMING: Encouraging Creativity

The brainstorming technique is widely used in industry and academia to encourage participants to generate ideas in an unhindered manner. In an academic context, brainstorming encourages students to participate actively in idea-generation exercises and experience benefits of a multi-dimensional approach to analyzing problems or solutions.

9. QUESTION-AND-ANSWER METHOD: Encouraging Student Participation

The goal of the question-and-answer method is to draw students into active participation in teaching and learning processes. The technique also encourages students to move beyond memorization to higher levels of learning that require clarification, expansion, generalization, and inference. Often students are conditioned to simply sit in class, take notes and then study and learn the information on their own. With the question-and-answer methodology, the students can learn in real-time, as they are being taught, which helps them understand and integrate the material better.

10. JOURNAL ARTICLE CRITIQUE

This strategy enables to the students to be aware of the developments in the field of their specialization and have the chance to write and express their opinions regarding the material. Continuous implementation of this kind of activity encourages the students to develop their own theories or philosophies that they may utilized in their conduct of research.

11. DIRECT METHODS

Direct methods involve the actual products of student work that can be evaluated in light of the learning outcomes. These may be the same materials for which students have been evaluated on an individual level (i.e., given a grade), but in this context one is looking across students at learning outcomes of groups. These methods could include:

a) Course-embedded assignments (e.g., tests, papers, reviews, presentations)

b) Standardized tests
c) Locally developed tests
d) Portfolio evaluation
e) Thesis or research paper
f) Oral exams
g) Focus groups

12. INDIRECT METHODS

Indirect methods are not based directly on academic or work performance. These usually involve perceptions by either the student or someone in a position to observe the student's work. These could include:

a) Student satisfaction surveys
b) Exit interviews
c) Focus groups
d) Alumni surveys
e) Employer surveys
f) External reviewer

XII. LABORATORY FACILITIES

LABORATORY FOR BIOINFORMATICS

Laboratory Software:

1.
2.
3.
4.
5.

Laboratory Equipments

1.
2.
3.

Laboratory Experiments

1.
2.
3.
4.
5.
6.
7.
8.

XIII. PROGRAM REFERENCES

1. Three reliable Syllabi of the three universities listed below as compared to Syllabi of Future University:

-Arizona University (U.S.A)

-Middle East Tennessee University (U.S.A)

-Glasco University (UX)

2. The standardization of international corporations:

-ACM (American computer Machinery)

-I.E.E.E (Industrial Electronics and Electrical Engineers)

3. The observation of external examiners considered from Ministry of Higher Education.

4. The feedback of the Lecturers of Bioinformatics Department.