This schedule is tentative. It will change according to instructors and their subject of expertise.

Course	Term Dates	Class Dates
Fall 2018	8/27/18-12/14/18	Friday/Saturday
Innovation and Technology (BUS 670)	8/27/18-12/10/18	9/7-8 & 10/5-6
Project Management (BUS 671)	10/22/18-12/7/18	11/2-3 & 11/30-12/1
Spring 2019		
Data Analysis and Analytics (CSC-500)		
Design and Analysis of Algorithms (CSC-550)		
Summer 2019		
Foundations of Artificial Intelligence (CSC-530)		
Secure Systems (CSC-540)		
Fall 2019		
Database Systems Principles(CSC-520)		
Programming Languages (CSC-570)		
Spring 2020		
Masters Project 1(CSC-600)		
Operating Systems (CSC-555)		
Summer 2020		
Masters Project 2 (CSC-601)		
Distributed Systems(CSC-560)		

WILLIAM JESSUP UNIVERSITY

Computer Science Major – Course Syllabus

CSCI-560: "Distributed Systems"

(3 units)

Instructor   Prerequisites

TBD

Course Description

Provides the examination of various components of software infrastructure required for distributing computing. Examines network protocols and API supporting this paradigm. Examines the necessary requirements of fault-tolerance, synchronization, consistency and security. Examination of some current distributed systems covering their architecture, design and implementation. Cloud computing systems will be examined. Understand the different synchronization and coordination techniques.

Course Objectives/Outcomes

Upon completion of the course, students will have learned:

- how to make use of the different industry distributed or cloud systems
- Able to design an application coordinating among the different distributed systems

Textbook

• None specified.

Supplemental Internet Resources

The Internet may be used to find additional resources. Course participants are encouraged to alert others in the class when a true gem is found. This is best accomplished through the use of the *course e-mail or other online tool* (TBD). Please take the time to check regularly (at least weekly) for new posts.

Course Content

Network communication and remote procedure calls.
DNS and content delivery networks.
Consistency and locking.
Fault tolerance - detection and correction.
Distributed mutual Exclusion.
Distributed File Systems.
Data intensive computing and Map/Reduce with Hadoop.
Time and synchronization.
Distributed Replication.
Examples of cloud systems from Google and Amazon.

WILLIAM JESSUP UNIVERSITY

CSCI-551: "Programming Languages"

(3 units)

Instructor	Prerequisites
TBD	

Course Description

Provides the fundamental concepts of programming languages. Introduction to functional languages like scheme, and functional aspects in procedural languages like Python. Introduction to language interpretation and a possible implementation of a language parser.

Course Objectives/Outcomes

Upon completion of the course, students will have learned:

- the student will be to ascertain different aspect and concepts of programming languages
- Understand a functional language like Scheme and see its effect on modern languages
- Able to understand language parsing and interpretation
- Able to construct a simple interpreter

Textbook

- None

Supplemental Internet Resources

The Internet may be used to find additional resources. Course participants are encouraged to alert others in the class when a true gem is found. This is best accomplished through the use of the *course e-mail or other online tool* (TBD). Please take the time to check regularly (at least weekly) for new posts.

Course Content

Evolution of the major programming languages
Syntax and Semantics of languages
Lexical and Syntax Analysis.
Programming language features like names, binding and scopes.
Data types
Abstract Data types
Support for Object-Oriented Programming.
Functional programming using scheme.
Functional aspects in Python.
Exception handling.

WILLIAM JESSUP UNIVERSITY
Computer Science Major – Course Syllabus

CSCI-561: "Design and Analysis of Algorithms"

(3 units)

Instructor
TBD

Prerequisites

Course Description
Provides the core principles of computer science: algorithmic thinking and computational problem solving. Techniques for the design and analysis of efficient algorithms, emphasizing methods useful in practice. Topics include divide-and-conquer, randomization, dynamic programming, incremental improvement, complexity, and cryptography.
[Analysis of Algorithms]

Course Objectives/Outcomes
Upon completion of the course, students will have learned:

- different data structures and associated algorithms
- performance of algorithms and how to analyze them
- various techniques in deciding the right algorithms for problems

Textbook
• none

Supplemental Internet Resources
The Internet may be used to find additional resources. Course participants are encouraged to alert others in the class when a true gem is found. This is best accomplished through the use of the *course e-mail or other online tool* (TBD). Please take the time to check regularly (at least weekly) for new posts.

Course Content

Review of asymptotics, basic data structures and algorithms.
Techniques in sorting and searching

- classical sorting, Interpolation search, Deterministic Kth selection
- Lower bounds on max & min, Majority detection

Advanced data structures

- skip lists, Amortized analysis, Fibonacci heaps

Graph Algorithms

- lowest common ancestor, Minimum spanning tree
- Minimum matchings, Degree constrained trees

Distributed Algorithms

- Distributed models
- Leader election

NP-completeness

- Polynomial time and intractability
- Space and time complexity
- Approximate heuristics

WILLIAM JESSUP UNIVERSITY
Computer Science Major – Course Syllabus
CSCI-555: "Operating Systems"
(3 units)

Instructor | Prerequisites

TBD

Course Description

The course provides a study of operating systems with all its abstraction and management design and policies relating to resource assignments. It will make use of Python and C for hands-on interaction with the operating system, especially on multithreading, multiprocessing and concurrency control. Particular emphasis will be given to three major OS subsystems: process management (processes, threads, CPU scheduling, synchronization, and deadlock), memory management (segmentation, paging, swapping), and file systems. It covers advanced topics in concurrency, deadlock protection, multiprocessor scheduling, computer system modeling, and virtual memory management from the operating systems viewpoint.

Course Objectives/Outcomes

Upon completion of the course, students will have learned:

- how to improve performance with the use of multithreading and multiprocessing
- how to affect the performance of programs with efficient use of the operating systems
- how to interact with multiple operating systems
- how to write efficient programs

Textbook

• none

Supplemental Internet Resources

The Internet may be used to find additional resources. Course participants are encouraged to alert others in the class when a true gem is found. This is best accomplished through the use of the *course e-mail or other online tool* (TBD)**.** Please take the time to check regularly (at least weekly) for new posts.

Course Content

Introduction to mobile computing and mobile application development.
Setup of the android development environment.
Discussion of three perspectives: mobile technology, application development and user interaction.
Common paradigms - low power computing, and limited resource environment.
Mobile computing in an environment with low resources, fault tolerance and persistence.
Mobile application frameworks and development environments.
Current research in mobile computing including wearables.

WILLIAM JESSUP UNIVERSITY

Computer Science Major – Course Syllabus

CSCI-571: "Foundations of Artificial Intelligence"

(3 units)

Instructor

TBD

Prerequisites

Course Description

Provides a study of the field of artificial intelligence in the areas associated with machine learning, generative modeling, network simulation, language processing and object recognition. Provides a survey of the various software architectures and projects related to the use and application of artificial intelligence.

This course builds upon the course on Introduction to Artificial Intelligence.

Course Objectives/Outcomes

Upon completion of the course, students will have learned:

- the current applications of artificial intelligence
- the current issues of artificial intelligence
- various topics within machine learning
- able to build an application based on AI techniques

Textbook

- none

Supplemental Internet Resources

The Internet may be used to find additional resources. Course participants are encouraged to alert others in the class when a true gem is found. This is best accomplished through the use of the *course e-mail or other online tool* (TBD). Please take the time to check regularly (at least weekly) for new posts.

Course Content

Introduction to classical AI (artificial intelligence) as well as non-classical AI.
Fundamental conditions, problems and challenges for AI also from a philosophical perspective.
Introduction to Google library of AI.
Natural Language Processing - classical and use of Google library.
Fundamentals of search: problem, solution, state space, breadth-first, depth-first, heuristics, A*, local search and optimization.
Knowledge representation: logic as form of expression (syntax and semantics of propositional logic and predicate logic).
Agent paradigms: the hierarchical paradigm, the reactive paradigm, and the hybrid paradigm.
Classical planning and execution.
The physical structure of robots. Teleoperation and semi-autonomous robots.
Embodied cognition and situatedness.
Neural networks: background and fundamentals.
Artificial evolution, genetic algorithms
Multiple autonomous agents, swarm intelligence, stigmergy, emergence.
Learning and simulation.

WILLIAM JESSUP UNIVERSITY
Computer Science Major – Course Syllabus
CSCI-565: "Data Analysis and Analytics**"**
(3 units)

Instructor	Prerequisites
TBD	

Course Description

This course provides the techniques and algorithms in the examination of data for meaningful patterns. Examination of the combination of mathematics, statistics, programming, and the context of the problem to come up with different insights. Examination into the management issue relating to the gathering and cleansing of data. The usage of both R and Python to look at particular groups of data, and to examine the different patterns. Familiarize student with most important technology information.

Course Objectives/Outcomes

- Understand the context of the problem
- Understand the tools and techniques required to solve the problem
- Understand the algorithms involved in doing the analysis
- Understand the problems associated with dirty and clean data

Textbook (Instructor will select one among the following)

- none

Supplemental Internet Resources

The Internet may be used to find additional resources. Course participants are encouraged to alert others in the class when a true gem is found. This is best accomplished through the use of the *course e-mail or other online tool* (TBD). Please take the time to check regularly (at least weekly) for new posts.

Course Content

Examine the basic tools for statistical analysis, R and Python, and the computing environment.
Examine several machine learning algorithms.
Use of Spark 2.0 which is a important big data processing framework.
Examine Spark ML (Machine Learning) API and Spark Streaming which allows analysis of data in flight, i.e. in near real time.
Examine NoSQL storage solutions exemplified by Cassandra for their critical features: speed of reads and writes, and ability to scale to extreme volumes.
Examine memory resident databases (VoltDB, SciDB) and graph databases (Ne4J).
Students will gain the ability to initiate and design highly scalable systems that can accept, store, and analyze large volumes of unstructured data in batch mode and/or real time.
Most lectures will be presented using Python examples.
Use of Google AI library.

WILLIAM JESSUP UNIVERSITY
Computer Science Major – Course Syllabus

CSCI-580: "Database Systems Principles"

(3 units)

Instructor
TBD

Prerequisites

Course Description

Provides the concepts and theory of database management systems. Topics include database system architectures, data models, query languages, conceptual and logical database design, physical organization, and transaction management. The entity-relationship model, relational model and object models are investigated in detail. Provision of object-relational databases. Considerations of implementation techniques of databases are examined.. NoSQL and object-oriented databases are examined.

Course Objectives/Outcomes

- Able to do SQL on most relational databases
- Able to perform optimization of most sql queries
- Able to interface with noSQL databases and know they work

Textbook (Instructor will select one among the following)

• none

Supplemental Internet Resources

The Internet may be used to find additional resources. Course participants are encouraged to alert others in the class when a true gem is found. This is best accomplished through the use of the *course e-mail or other online tool* (TBD). Please take the time to check regularly (at least weekly) for new posts.

Course Content

Examine query within the context of joins, subqueries and set theoretic queries
Write stored procedures and queries within object relational database like postgreSQL.
Apply the principles of query optimization to a database schema.
Locking and optimistic locking within relational database.
Examine the object-relational aspects of postgresQL.
Perform queries against object-relational database.
Examine the noSQL database like MongoDB and Hadoop.
Apply and customize state-of-the-art implementation techniques for single-node database management systems following modern coding practices.
Identify trade-offs among database systems techniques and contrast alternatives for both on-line transaction processing and on-line analytical workloads. Interpret and comparatively criticize state-of-the-art research talks and papers, with emphasis on constructive improvements.

WILLIAM JESSUP UNIVERSITY
Computer Science Major – Course Syllabus
CSCI-535: "Secure Systems"
(3 units)

Instructor
TBD

Prerequisite

Course Description
Covers the principles and operations of operating systems, distributed operating systems and network systems in the context of security. Examine network security protocols looking at its strengths and weaknesses. Covers the use of network and OS tools for system management and security management.
[Operating Systems and Computer Networking]

Course Objectives/Outcomes

- Able to recognize the different holes and vulnerabilities in the design of applications or systems
- Able to design tests to find these holes
- Able to figure out how to design security to eliminate these security holes

Textbook (Instructor will select one among the following)
• none

Supplemental Internet Resources

The Internet may be used to find additional resources. Course participants are encouraged to alert others in the class when a true gem is found. This is best accomplished through the use of the *course e-mail or other online tool* (TBD). Please take the time to check regularly (at least weekly) for new posts.

Course Content

Explore the building blocks needed to implement a life-cycle security system.
Instruction focuses on how to analyze internal applications, computing platforms/network infrastructure, and corporate objectives with an eye toward designing flexible security architecture that is best suited for the enterprise including how to define security architecture and what security architecture describes
Design and implementation of secure computer systems.
Discuss the threat models, attacks that compromise security, and techniques for achieving security, based on recent research papers.
Topics include operating system (OS) security, capabilities, information flow control, language security, network protocols, hardware security, and security in web applications.
Assignments involve implementing and compromising a secure web server and web application, and a group final project.
Case studies are used to illustrate key security architecture concepts and methods.
Study of some open-source secure management system.