Coding Technology
Mississippi Curriculum Framework

CIP: 11.0202 Computer Programming, Specific Applications.

2018

Published by:
Mississippi Community College Board
Division of Workforce, Career, and Technical Education
3825 Ridgewood Road
Jackson, MS 39211
Phone: 601-432-6155
Email: curriculum@mccb.edu

FACULTY WRITING TEAM MEMBERS

Curtis Beckman, Program Coordinator Computer Programming Technology, Meridian Community College
Terry Cullum, Computer Networking Technology Instructor, Meridian Community College
Natasha Lewis, IST Instructor, Northeast Mississippi Community College
Brandon Sesser, Information Systems Technology Director/Cyber Security Instructor, East Mississippi Community College
David Rose, Information Systems Technology, District Coordinator and Lead Online Instructor, Hinds Community College
Jo Ponder, Instructor Computer Programming Technology, Hinds Community College

ADMINISTRATOR WRITING TEAM MEMBERS

Dr. David Campbell, Vice President for Workforce Solutions and Career-Technical Education, Northwest Mississippi Community College
Brent Duguid, Dean of Career Technical & Workforce Education, Copiah Lincoln Community College
Sherry Franklin, Associate Vice President for Career and Technical Education and District Director of Career and Technical Education and Rankin Campus Career and Technical Dean, Hinds Community College
Charles J. Cook, Assistant Dean of Career and Technical Education & Workforce, East Mississippi Community College
Jason Mattox, Associate Vice President Career and Technical Education, Northeast Mississippi Community College
Michelle Bargona, Vice President of Instruction, Northeast Mississippi Community College
Dr. Richie McAllister, Associate Vice President for Workforce Education, Meridian Community College
Barry Emison, Dean of Career and Technical Education, Itawamba Community College
John Shows, Associate Vice President of Community Campus and Career/Technical Education, Mississippi Gulf Coast Community College
Rod Tolbert, Dean of Career and Technical Education, Jones College
Dr. Raj K. Shaunak, Vice President Workforce and Community Services, East Mississippi Community College

BUSINESS AND INDUSTRY CONTRIBUTING TEAM MEMBERS

*Sean Anthony, Executive Director, Base Camp Coding Academy
*Nate Clark, Senior Technical Director, Basecamp Coding Academy
* John Marsalis, Vice President, Software Engineering, CoreLogic
*Carla Lewis, Senior Vice President of IT, C Spire
*Shelly Hollis, Assistant Director RCU, Center for Cyber Education, Mississippi State University
*Corey Mize, Computer Scientist

***Denotes an industry member who attended the writing team meeting and/or participated in reviewing the curriculum content.**

OFFICE OF CURRICULUM AND INSTRUCTION TEAM MEMBERS

Scott Kolle, Ph.D., Director of Curriculum and Instruction, Mississippi Community College Board
LaToya Sterling Ph.D., Curriculum Specialist, Office of Curriculum and Instruction, Mississippi Community College Board
Sheriece Robinson Ed.D, Curriculum Specialist, Office of Curriculum and Instruction, Mississippi Community College Board

The Office of Curriculum and Instruction (OCI) was founded in 2013 under the Division of Workforce, Career, and Technical Education at the Mississippi Community College Board (MCCB). The office is funded through a partnership with The Mississippi Department of Education (MDE), who serves as Mississippi's fiscal agent for state and federal Career and Technical Education (CTE) Funds. The OCI is tasked with developing statewide CTE curriculum, programming, and professional development designed to meet the local and statewide economic demand.

For information, please contact curriculum@mccb.edu

Contents

NATIONAL CERTIFICATION STANDARDS

The Coding Technology content is aligned to the introductory Microsoft certification for individuals considering a career in technology. These certifications addresses a wide spectrum of fundamental technical concepts, assesses and validates your core technical knowledge, and enhances your technical credibility.
The Coding Technology content is aligned to:
Microsoft Exam 98-383 Introduction to Programming Using HTML and CSS
Microsoft Exam 98-364 Database Fundamentals
Microsoft Exam 98-381 Introduction to Programming Using Python
Microsoft Exam 98-382 Introduction to Programming Using JavaScript

For more information related to Microsoft Technology Associate please visit https://www.microsoft.com/en-us/learning/mta-certification.aspx

INDUSTRY JOB PROJECTION DATA

The software developers (applications and software) occupations are projected to grow at 26.46% statewide. Median annual income for this occupation is $80, 629.12 at a state level. A summary of occupational data from the State Workforce Investment Board Data Center is displayed below:

Table 1: Education Level

Program Occupations	**Education Level**
Coding/software developers, applications and software	*High school or higher with specialized training related to the subject matter **Bachelor's degree

*Current trend in Mississippi is less than a Bachelor's degree for some entry level jobs within the state and nationally. Jobs outside of the state may allow for the employee to work remotely from the State of Mississippi.

** Job data for Tables 1, 2, and 3 are based on Bachelor's degree at a National level.

Table 2: Occupational Overview**

	Region	**State**	**United States**
2014 Occupational Jobs	4,581	4,581	1,233,163
2024 Occupational Jobs	5,793	5,793	1,870,980
Total Change	1,212	1,212	637,817
Total % Change	26.46%	26.46%	51.72%
2014 Median Hourly Earnings	$38.76	$38.76	$49.75
2014 Median Annual Earnings	***$66,580.80 ****$80,629.12	***$66,580.80 ****$80,629.12	***$79,851.20 ****$103,480.00
Annual Openings	121	121	63782

** Job data for Tables 1, 2, and 3 are based on Bachelor's degree at a National level.

*** Coding and Software Coding

***Software developers (applications and software)

Table 3: Occupational Breakdown**

Description	**2014 Jobs**	**2024 Jobs**	**Annual Openings**	**2016 Hourly Earnings**	**2016 Annual Earnings 2,080 Work Hours**
Software Developers, Applications	3,045	4,155	111	$40.57	$84,385.60
Software Developers, Systems Software	1,536	1,638	10	$36.70	$76,336.00
Total	4,581	5,793	121	$38.76	$80,620.80

** Job data for Tables 1, 2, and 3 are based on Bachelor's degree at a National level.

Table 4: Occupational Change**

Description	**Regional Change**	**Regional % Change**	**State % Change**	**National % Change**
Software Developers, Applications	1,110	36.45%	36.45%	69.90%
Software Developers, Systems Software	102	6.64%	6.64%	15.86%

** Job data for Tables 1, 2, and 3 are based on Bachelor's degree at a National level.

Articulation

None at this time.

Technical Skills Assessment

Colleges should report the following for students who complete the program with a career certificate, technical certificate, or an Associate of Applied Science Degree for technical skills attainment. To use the approved Alternate Assessment for the following programs of study, colleges should provide a Letter of Notification to the Director of Career Technical Education at the MS Community College Board. Please see the following link for further instructions: http://www.mccb.edu/wkfEdu/CTDefault.aspx.

CIP Code	**Program of Study**	
11.0202	Coding Technology	
Level	**Standard Assessment**	**Alternate Assessment EPT**
College Credit Certificate (15)	Microsoft Exam 98-383 Introduction to Programming Using HTML and CSS*	
Career (30)	Microsoft Exam 98-382 Introduction to Programming Using JavaScript* Microsoft Exam 98-364 Database Fundamentals*	
Level	Standard Assessment	**Alternate Assessment EPT**
Technical/AAS	Microsoft Exam 98-381 Introduction to Programming Using Python*	

***Certification schedule can be adjusted pending course offerings.**

RESEARCH ABSTRACT

In the fall of 2018 and the spring of 2019, the Office of Curriculum and Instruction (OCI) met with the different industry members, State agencies, and community colleges in an effort to give insight on the importance and necessity of this program. The committee members collaborated together in determining the structure of the program and its courses. An industry questionnaire was used to gather feedback concerning the trends and needs, both current and future, of their field. Program faculty, administrators, and industry members were consulted regarding industry workforce needs and trends.

Industry advisory team members from the college involved with this program were asked to give input related to the creation of this curriculum framework.

A curriculum writing meeting was convened in February 2019 and attended by faculty, administrators, and industry members. During the writing meeting, the curriculum was aligned to the 30/45/60 model allowing for a stackable credentials model whereby Students can earn a career certificate, technical certificate, and an Associate of Applied Science degree.

REVISION HISTORY:

2019 Mississippi Community College Board

PROGRAM DESCRIPTION

The computer coding/software development curriculum is a two-year program of study designed to train students to be software developers. Students will work with real world technologies to learn the fundamentals of coding, app development, web applications, and the life leadership skills needed to be successful in their careers and competitive in the job market. The Coding Technology curricu.um is aligned to a skillset that enables students to learn and enhance skills as they relate to: HTML and CSS, Database Fundamentals, Python, and JavaScript.

Suggested Course Sequence-Coding

Accelerated Integrated Career Pathway

			SCH Breakdown			Contact Hour Breakdown		Certification Information
Course Number	Course Name	Semester Credit Hours	Lecture	Lab	Total Contact Hours	Lecture	Lab	Certification Name
IST 1724	Programming in Python	4	2	4	90			
IST 1513	SQL Programming	3	2	2	60			
IST 1154	Web & Programming Concepts	4	2	4	90			
IST 1414	Client-Side Programming	4	2	4	90			
	Total	**15**						

Career Certificate Required Courses

			SCH Breakdown			Contact Hour Breakdown		Certification Information
Course Number	Course Name	Semester Credit Hours	Lecture	Lab	Total Contact Hours	Lecture	Lab	Certification Name
IST 1724	Programming in Python	4	2	4	90			
IST 1513	SQL Programming	3	2	2	60			
IST 1154	Web & Programming Concepts	4	2	4	90			
IST 1414	Client-Side Programming	4	2	4	90			
	Electives	15						
	Total	**30**						

Technical Certificate Required Courses

			SCH Breakdown			Contact Hour Breakdown		Certification Information
Course Number	Course Name	Semester Credit Hours	Lecture	Lab	Total Contact Hours	Lecture	Lab	Certification Name
IST 2724	Adv. Java Programming	4	2	4	90			
	Electives	11						
	Total	**15**						

Associate Degree Required Courses

			SCH Breakdown			Contact Hour Breakdown		Certification Information
Course Number	Course Name	Semester Credit Hours	Lecture	Lab	Total Contact Hours	Lecture	Lab	Certification Name
	Humanities/Fine Arts	3						
	Social/Behavioral Sciences	3						
	Math/Science	3						
	Academic electives	6						

General Education Core Courses

General Education Core Courses

To receive the Associate of Applied Science degree, a student must complete all of the required coursework found in the Career Certificate option, Technical Certificate option, and a minimum of 15 semester hours of General Education core. The courses in the General Education Core may be spaced out over the entire length of the program so that students complete some academic and Career Technical courses each semester or provided primarily within the last semester. Each community college will specify the actual courses that are required to meet the General Education Core Requirements for the Associate of Applied Science degree at their college. The Southern Association of Colleges and Schools Commission on Colleges (SACSCOC) Section 9 Standard 3 of the *Principles of Accreditation: Foundations for Quality Enhancement*[1] describes the general education core.

Section 9 Standard 3:

3. The institution requires the successful completion of a general education component at the undergraduate level that

   a) is based on a coherent rationale.

   b) is a substantial component of each undergraduate degree program. For degree completion in associate programs, the component constitutes a minimum of 15 semester hours of the equivalent; for baccalaureate programs, a minimum of 30 semester hours or the equivalent.

   c) ensures breadth of knowledge. These credit hours include at least one course from each of the following areas: humanities/fine arts, social/behavioral sciences, and natural science/mathematics. These courses do not narrowly focus on those skills, techniques, and procedures specific to a particular occupation or profession.

<<<Add any additional general education standards as required for programmatic accreditation here and footnote below.>>>

General Education Courses

			SCH Breakdown			Contact Hour Breakdown		Certification Information
Course Number	Course Name	Semester Credit Hours	Lecture	Lab	Total Contact Hours	Lecture	Lab	Certification Name
	Humanities/Fine Arts	3						
	Social/Behavioral Sciences	3						
	Math/Science	3						
	Academic electives	6						

[1] Southern Association of Colleges and Schools Commission on Colleges. (2017). *The Principles of Accreditation: Foundations for Quality Enhancement*. Retrieved from http://www.sacscoc.org/2017ProposedPrinc/Proposed%20Principles%20Adopted%20by%20BOT.pdf

Elective Courses

			SCH Breakdown				Clock Hour Breakdown		
Course Number	Course Name	Semester Credit Hours	Lecture	Lab	Clinical/ Internship	Total Clock Hours	Lecture	Lab	Clinical/ Internship
IST 1523	SQL Programming II								
IST 1764	Programming in Python II								
IST 1714	Java Programming								
IST 2814	Full-Stack Web Development								
IST 2884	Full-Stack Web Application								
IST 2954	Capstone Coding Project								
SSP 1002	Smart Start								
Any other elective that supports the students pathway and approved by the Instructor									

Coding Courses

Course Number and Name: **IST 1154 Web and Programming Concepts**

Description: This course is an introduction to Web site development and programming logic. Students will gain hands-on experience in the development of computer programs. Upon completion of this course, students will be able to create a website.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
4	2	4	90

Prerequisite: None

Student Learning Outcomes:

1. Design a Web page using HTML and/or XHTML.
   a. Explain Web page creation.
   b. Contrast text editors and GUI editors.
   c. Explain HTML and XHTML.
      (1) Describe HTML, XHTML, and the standards of each.
      (2) Discuss Web browsers and standards.
      (3) Discuss Web page accessibility.
   d. Explain HTML and/or XHTML coding.
      (1) Explain markup tags.
      (2) Discuss and use document structure tags.
      (3) Demonstrate and use paragraph formatting, block-level elements, and text level elements.
      (4) Describe and code lists.
      (5) Use comments and good coding practices.
   e. Demonstrate graphical elements.
      (1) Explain and use the horizontal rule.
      (2) Discuss images; use them in a Web page.
      (3) Discuss image file formats; explain image optimization.
      (4) Discuss colors and the Web-safe color palette; implement them.
      (5) Explain special characters.
      (6) Use page colors and backgrounds.
      (7) Incorporate fonts into a Web page.
   f. Describe the use of hyperlinks.
      (1) Explain and code hyperlinks.
      (2) Use image and internal links.
   g. Explain the use of tables.
      (1) Demonstrate tables.
      (2) Use table and data alignment options.
      (3) Explain and use column and row spanning.
   h. Discuss forms.
      (1) Describe form uses and form fields.
      (2) Create forms.
   i. Evaluate image techniques.
      (1) Create image maps.
      (2) Discuss and use image transparency.
      (3) Explain interlacing.
      (4) Demonstrate the use of animated GIF images.
   j. Demonstrate Web page layout and elements.
      (1) Demonstrate effective layout.

(2) Discuss color and Web design.
(3) Explore font usage.
(4) Explain Web site usability testing.

k. Demonstrate navigation concepts.
(1) Explain the elements of navigation design.
(2) Demonstrate primary and secondary navigation.
(3) Discuss navigation hierarchy.
(4) Discuss site structure, uniform resource locators (URLs), and file names.
(5) Discuss familiar navigation conventions.

l. Research the standards organizations.
(1) Demonstrate Web site usability Explain the Internet governing bodies.
(2) Discuss the Internet Society (ISOC) and Internet Architecture Board (IAB).
(3) Describe the Internet Research Task Force (IRTF) and Internet Engineering Task Force (IETF).
(4) Discuss the World Wide Web Consortium (W3C), Names and Numbers (ICANN), and Requests for Comments (RFCs).

m. Demonstrate Web site usability testing

2. Demonstrate cascading style sheets

a. Use CSS language to build cascading style sheets.
(1) Discuss the history of CSS.
(2) Discuss basic CSS components.
(3) Discuss CSS rules.
(4) Build a basic style sheet.

b. Examine basic CSS techniques.
(1) Apply selection techniques.
(2) Apply effective font usage.
(3) Apply the CSS box model.
(4) Use color in CSS.

c. Apply advanced CSS techniques.
(1) Apply tables.
(2) Apply lists.
(3) Apply positioning elements.

3. Use program design tools.

a. Demonstrate the use of a flowchart.
b. Develop pseudo code.
c. Generate a hierarchy chart.

4. Discuss structured or modular programming.

a. Explain sequencing.
b. Explain selection.
c. Explain iteration.

5. Describe the philosophy of object-oriented programming.

a. Demonstrate event-driven programming.
b. Explain procedures.
c. Examine classes.

6. Create applications using program development steps.

a. Discuss the steps in the program development life cycle.
b. Design a program.
c. Code a program.

d. Test and debug a program

Standards:

Microsoft Technology Associate (MTA) Software Development Exam Objectives

- Understand web page development
- Understand Microsoft ASP.NET web application development
- Understanding web hosting
- Understanding web services

DRAFT

Course Number and Name: **IST 1414 Client –Side Programming**

Description: This course offers a comprehensive understanding of programming using JavaScript.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
4	2	4	90

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Demonstrate client-side programming using JavaScript.
    a. Apply JavaScript fundamentals.
        (1) Explain scripting languages.
        (2) Discuss JavaScript versus other languages.
        (3) Discuss client-side versus server-side programming.
        (4) Demonstrate embedding in HTML.
    b. Discuss JavaScript variables and data.
        (1) Demonstrate communicating with the user.
        (2) Explain variables and keywords.
        (3) Discuss expressions and operators.
        (4) Discuss inline scripting, simple user events, and the onLoad and onUnload event handlers.
    c. Explain functions, methods, and events.
        (1) Discuss functions.
        (2) Use defining and calling functions.
        (3) Use event handlers.
    d. Apply control structures and statements.
        (1) Apply decision statements.
        (2) Apply repetition statements.
    e. Examine the document object model (DOM).
        (1) Identify properties and methods of the window object.
        (2) Identify properties and methods of the location object.
        (3) Identify properties and methods of the history object.
        (4) Identify properties of the navigator object.
    f. Examine forms.
        (1) Discuss validation.
        (2) Validate form data.
    g. Discuss cookies and security.
        (1) Explain state information.
        (2) Describe query strings.
        (3) Save state information with cookie
    h. Debug JavaScript.
        (1) Use debugging tools.
        (2) Discuss debugging techniques.

DRAFT

Course Number and Name: **IST 1513 SQL Programming**

Description: This course is the first of a two-part series that offers students an extensive introduction to data server technology, covering the concepts of both relational and object relational databases and the structured query language (SQL). Students are taught to retrieve data and produce readable output.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
3	2	2	60

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Write basic SQL statements.
   a. Explore the capabilities of SQL SELECT statements.
   b. Execute basic SELECT statements.
   c. Use tools that recognize and submit SQL statements for processing.

2. Restrict and sort data.
   a. Limit the rows retrieved by a query.
   b. Sort the rows retrieved by a query.

3. Use single-row functions.
   a. Describe the various types of functions available in SQL.
   b. Use character, number, and date functions in SELECT statements.
   c. Demonstrate the use of conversion functions and conditional expressions.
   d. Use comparison operators and logical operators.
   e. Use range conditions, membership conditions, and pattern matching.
   f. Discuss rules of precedence.

4. Display data from multiple tables.
   a. Write SELECT statements to access data from more than one table.
   b. Describe the Cartesian product.
   c. Compare and use types of joins.

5. Aggregate data using group functions.
   a. Identify the available group functions.
   b. Discuss the uses of group functions.
   c. Demonstrate the grouping of data and include or exclude grouped rows.

6. Write subqueries.
   a. Identify the types of problems that subqueries can solve.
   b. Describe subqueries.
   c. List the types of subqueries.
   d. Write single-row and multiple-row subqueries.

7. Produce readable output.
   a. Produce queries requiring an input variable.
   b. Use tools to control environmental
   c. Create and execute script files.
8. Manipulate data in the database.

a. Describe data manipulation commands.
b. Insert rows into a table, update rows in a table, and delete rows from a table.
c. Control database transactions.
d. Discuss and implement read consistency.

9. Create and manage tables.
   a. Create tables.
   b. Describe the data types that can be used when specifying column definitions.
   c. Alter table definitions.
   d. Delete, rename, and truncate tables.

10. Use constraints.
    a. Describe constraints.
    b. Create and maintain constraints.

11. Create views.
    a. Describe views and their uses.
    b. Demonstrate how to create and delete a view.
    c. Retrieve data through a view.
    d. Alter the definition of a view.
    e. Insert, update, and delete data through a view.
    f. Discuss inline views.

12. Use other database objects in
    a. Manage database objects using a data dictionary.
    b. Create, maintain, and delete sequences.
    c. Create, maintain, and delete indexes.
    d. Create and delete private and public synonyms.

13. Control user access.
    a. Analyze the concepts of users and their roles and privileges.
    b. Practice creating users.
    c. Grant and revoke privileges.
    d. Create roles and grant privileges to roles.
    e. Change user passwords.

Standards:
CIW Database Design Specialist Exam Objectives
Domain 4:
4.1 Identify SQL commands and syntax
4.2 Create statements using Data Definition Language (DDL)
4.3 Form commands using Data Manipulation Language (DML)
4.4 Use Data Control Language (DCL) statements to control the access to data in a database and to grant users permission for data operations.

Course Number and Name: **IST 1523 SQL Programming II**

Description: This course is the first of a two-part series that offers students an extensive introduction to data server technology, covering the concepts of both relational and object relational databases and the structured query language (SQL). Students are taught to retrieve data and produce readable output.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
3	2	2	60

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Write basic SQL statements.
   a. Explore the capabilities of SQL SELECT statements.
   b. Execute basic SELECT statements.
   c. Use tools that recognize and submit SQL statements for processing.

2. Restrict and sort data.
   a. Limit the rows retrieved by a query.
   b. Sort the rows retrieved by a query.

3. Use single-row functions.
   a. Describe the various types of functions available in SQL.
   b. Use character, number, and date functions in SELECT statements.
   c. Demonstrate the use of conversion functions and conditional expressions.
   d. Use comparison operators and logical operators.
   e. Use range conditions, membership conditions, and pattern matching.
   f. Discuss rules of precedence.

4. Display data from multiple tables.
   a. Write SELECT statements to access data from more than one table.
   b. Describe the Cartesian product.
   c. Compare and use types of joins.

5. Aggregate data using group functions.
   a. Identify the available group functions.
   b. Discuss the uses of group functions.
   c. Demonstrate the grouping of data and include or exclude grouped rows.

6. Write subqueries.
   a. Identify the types of problems that subqueries can solve.
   b. Describe subqueries.
   c. List the types of subqueries.
   d. Write single-row and multiple-row subqueries.

7. Produce readable output.
   a. Produce queries requiring an input variable.
   b. Use tools to control environmental
   c. Create and execute script files.
8. Manipulate data in the database.

- a. Describe data manipulation commands.
- b. Insert rows into a table, update rows in a table, and delete rows from a table.
- c. Control database transactions.
- d. Discuss and implement read consistency.

9. Create and manage tables.
    - a. Create tables.
    - b. Describe the data types that can be used when specifying column definitions.
    - c. Alter table definitions.
    - d. Delete, rename, and truncate tables.

10. Use constraints.
    - a. Describe constraints.
    - b. Create and maintain constraints.

11. Create views.
    - a. Describe views and their uses.
    - b. Demonstrate how to create and delete a view.
    - c. Retrieve data through a view.
    - d. Alter the definition of a view.
    - e. Insert, update, and delete data through a view.
    - f. Discuss inline views.

12. Use other database objects in
    - a. Manage database objects using a data dictionary.
    - b. Create, maintain, and delete sequences.
    - c. Create, maintain, and delete indexes.
    - d. Create and delete private and public synonyms.

13. Control user access.
    - a. Analyze the concepts of users and their roles and privileges.
    - b. Practice creating users.
    - c. Grant and revoke privileges.
    - d. Create roles and grant privileges to roles.
    - e. Change user passwords.

Course Number and Name: **IST 1714 Java Programming Language**

Description: This introduction to the Java programming language is to include sort, loops, and arrays.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
4	2	4	90

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Develop programs that use arithmetical operations and logical operations.
   a. Use variables, constants, and strings.
   b. Use methods, classes, and objects.

2. Develop programs using decision making and conditional breaks.
   a. Develop programs that utilize the “if and if...else” structure.
   b. Develop programs that utilize the nested If structure.

3. Use looping structures in programs.
   a. Develop programs that implement “while” and “do while” loops.
   b. Develop programs that implement “for the” loop.
   c. Develop programs that implement nested loops.

4. Use arrays.
   a. Develop programs that require defining, loading, and searching an array.
   b. Develop programs that sort an array.

Course Number and Name: **IST 1724 Programming in Python**

Description This course is designed to provide an introduction to programming concepts and data informatics using Python through lecture and a series of practical hands-on exercises.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
3	2	2	60

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Describe proper programming techniques.
   a. Discuss Python terminology
   b. Explore the building blocks of programs
   c. Develop algorithms

2. Using fundamental Python concepts.
   a. Declare variables and determine data types
   b. Create expressions while following the proper order of operations
   c. Manipulate strings
   d. Add proper comments for documentation
   e. Request user input and process output
   f. Debug python programs

3. Execute conditional structures to control program flow
   a. Use Boolean expressions and logical operators
   b. Incorporate decision structures and loops in programs

4. Design programs using functions
   a. Use functions call and built in functions
   b. Provide parameters and arguments as required by functions

5. Process data
   a. Open, read, search and write to files
   b. Use lists, tuples and strings
   c. Discuss dictionaries and sets

6. Discuss object-oriented programming
   a. Develop programs with classes
   b. Discuss inheritance and recursion

DRAFT

Course Number and Name: **IST 1764 Programming in Python II**

Description This course is designed to provide advanced information to programming concepts and data informatics using Python through lecture and a series of practical hands-on exercises.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
4	2	4	90

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Develop algorithms
    a. Searching for the 2 smallest values
    b. Timing the functions
    c. At a minimum, you saw this
    d. Exercises
2. Searching and Sorting excerpt
    a. Searching a list
    b. Binary Search
    c. Sorting
    d. More Efficient Sorting Algorithms
    e. Merge Sort: A Faster Sorting Algorithm
    f. Sorting Out What You Learned
    g. Exercises
3. Object-Oriented Programming
    a. Understanding a Problem Domain
    b. Function isinstance, Class object, and Class Book
    c. Writing a Method in Class Book
    d. Plugging into Python Syntax: More Special Methods
    e. A Little Bit of 00 Theory
    f. A Case Study: Molecules, Atoms, and PDB Files
    g. Classifying What You've Learned
    h. Exercises
4. Testing and Debugging
    a. Why Do You Need to Test?
    b. Case Study: Testing above freezing
    c. Case Study: Testing running sum
    d. Choosing Test Cases
    e. Hunting Bugs
    f. Bugs We've Put in Your Ear
    g. Exercises
5. Creating Graphical User Interfaces
    a. Using Module tinker
    b. Building a Basic Gill
    c. Models, Views, and Controllers, Oh My!
    d. Customizing the Visual Style
    e. Introducing a Few More Widgets
    f. Object-Oriented Gills

   g. Keeping the Concepts from Being a GUI Mess
   h. Exercises

6. Databases
   a. Overview
   b. Creating and Populating
   c. Retrieving Data
   d. Updating and Deleting
   e. Using NULL for Missing Data
   f. Using Joins to Combine Tables
   g. Keys and Constraints
   h. Advanced Features
   i. Some Data Based On What You Learned
   j. Exercises

Course Number and Name: **IST 2724 Advanced Java Programming**

Description: This course is a second of a two-part series that offers students an extensive introduction into Java Programming. Students will be taught advanced concepts of arrays, inheritance, applets, and swing components.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
4	2	4	90

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Use multidimensional arrays
   a. Use two-Dimensional arrays in programs
   b. Demonstrate the use of different Array classes

2. Develop Programs using advance Inheritance concepts
   a. Use Extend classes in programs
   b. Use override superclasses in programs
   c. Understand and employ information hiding in programs
   d. Create and use Abstract Classes
   e. Use Arrays of subclass objects

3. Process File input and Output
   a. Open, read, search and write to files
   b. Discuss random access data files

4. Demonstrate and understand Swing Components
   a. Use the JFrame and JLabel in programs
   b. Discuss event-driven programming
   c. Demonstrate Using GUI based classes

Course Number and Name: **IST 2814 Full-Stack Web Development**

Description: This course offers students an introduction into Full-Stack Web Development (Django or other applicable software). Students will be taught concepts related to Django and other aspects of fullstack web development components and applications.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
4	2	4	90

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Basics of Django or other applicable software and the model template View
2. Planning and Construction Models
3. Mapping Data Structures
4. Site Administration
5. URL Routing
6. Creating Generic and Custom Views
7. Displaying Data and templates
8. Building Forms
9. End of course project: Full Stack web application managing a database of inventory for a fictional business.

Course Number and Name: **IST 2884 Full-Stack Web Application**

Description: This course is a second of a two-part series that offers students an extensive introduction into web application (Spring or other applicable software). Students will be taught advanced concepts of components.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
4	2	4	90

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Creating and building projects with Spring boot or other applicable software.
2. Creating controllers in Spring MVC or other applicable software.
3. Designing views in JSP.
4. Processing and validating form input.
5. Database management with JDBC.
6. Securing applications with Spring security or other applicable software.
7. End of course project: Spring or other applicable software web application Development: Operational full stack web application for customers of a fictional service provider (JAVA/Spring/SQL or other applicable software)

DRAFT

Course Number and Name: **IST 2954 Capstone Coding Project**

Description: This course is designed to encourage student to think critically, solve challenging issues. Students will apply the skills gained to a coding project and/projects.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Contact Hours
4	2	4	90

Prerequisite: Instructor approved

Student Learning Outcomes:

This project is designed to provide the student with an opportunity to work on a real world problem/project. Emphasis is placed on "real world" work experience as it integrates academics with practical applications that relate meaningfully to careers in the coding/software development. Students, with the instructor's approval, will be paired with an employer. The project will be based on employer's area of interest or need. Grades for this course will be based on a combination of the employer's evaluation of the student, and the contents of a report submitted by the student. The final results of the code project will published online in a code repository (employer permission needed).

Upon completion of this course, the student should be able to demonstrate the ability to apply knowledge and skills gained in the classroom to a "real world" work experience. Make a final presentation/demo part of the requirements. Using communication/presentation skills to explain your work.

Course Number and Name: **Work-Based Learning I, II, III, IV, V, and VI**

WBL 191(1-3), WBL 192(1-3), WBL 193(1-3), WBL 291(1-3), WBL 292(1-3), and WBL 293(1-3)

Description: A structured work-site learning experience in which the student, program area teacher, Work-Based Learning Coordinator, and worksite supervisor/mentor develop and implement an educational training agreement. Designed to integrate the student's academic and technical skills into a work environment. Includes regular meetings and seminars with school personnel for supplemental instruction and progress reviews.

Hour Breakdown:

Semester Credit Hours	Lecture	Lab	Externship	Contact Hours
1			3	45
2			6	90
3			9	135

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Apply technical skills and related academic knowledge needed to be a viable member of the workforce.
   a. Demonstrate technical skills necessary to complete job requirements.
   b. Demonstrate academic skills necessary to complete job requirements.
   c. Perform tasks detailed in an educational training agreement at the work setting.
   d. Demonstrate knowledge of employability skills such as creating cover letters, resumes, etc.

2. Apply general workplace skills to include positive work habits necessary for successful employment.
   a. Demonstrate appropriate human relationship skills in the work setting to include conflict resolution, team participation, leadership, negotiation, and customer/client service.
   b. Utilize time, materials, and resource management skills.
   c. Use critical thinking skills such as problem solving, decision making, and reasoning.
   d. Acquire, evaluate, organize, maintain, interpret, and communicate information.

Appendix A: Recommended Tools and Equipment

CAPITALIZED ITEMS

1. Computer or notebook work centers with multimedia support (camera, audio)
2. Peripherals and access:
    a. Color laser printer (1 per classroom)
    b. Laser printer (1 per lab)
    c. Scanner, color page (1 per lab)
3. Multimedia presentation system (1 per lab)
4. Teacher workstation or notebook with printer, scanner, and Internet access

NON-CAPITALIZED ITEMS

RECCOMMENDED INSTRUCTIONAL AIDES

1. Microsoft© Office software or equivalent productivity, latest version
2. Windows, latest version
3. Spring Framework (open source) with documentation (latest edition)
4. Django Framework (open source) with documentation (latest edition)

d. Internet access

Classroom set up: desk and chair (1 per student)

Appendix B: Curriculum Definitions and Terms

- Course Name – A common name that will be used by all community colleges in reporting Students

- Course Abbreviation – A common abbreviation that will be used by all community and junior colleges in reporting students

- Classification – Courses may be classified as the following:
    - Career Certificate Required Course – A required course for all students completing a career certificate.
    - Technical Certificate Required Course – A required course for all students completing a technical certificate.
    - Technical Elective – Elective courses that are available for colleges to offer to students.

- Description – A short narrative that includes the major purpose(s) of the course

- Prerequisites – A listing of any courses that must be taken prior to or on enrollment in the course

- Corequisites – A listing of courses that may be taken while enrolled in the course

- Student Learning Outcomes – A listing of the student outcomes (major concepts and performances) that will enable students to demonstrate mastery of these competencies

The following guidelines were used in developing the program(s) in this document and should be considered in compiling and revising course syllabi and daily lesson plans at the local level:

- The content of the courses in this document reflects approximately 75% of the time allocated to each course. The remaining 25% of each course should be developed at the local district level and may reflect the following:
    - Additional competencies and objectives within the course related to topics not found in the state framework, including activities related to specific needs of industries in the community college district
    - Activities that develop a higher level of mastery on the existing competencies and suggested objectives
    - Activities and instruction related to new technologies and concepts that were not prevalent at the time the current framework was developed or revised
    - Activities that include integration of academic and career–technical skills and course work, school-to-work transition activities, and articulation of secondary and postsecondary career–technical programs
    - Individualized learning activities, including work-site learning activities, to better prepare individuals in the courses for their chosen occupational areas

- Sequencing of the course within a program is left to the discretion of the local college. Naturally, foundation courses related to topics such as safety, tool and equip mental usage, and other fundamental skills should be taught first. Other courses related to specific skill areas and related academics, however, may be sequenced to take advantage of seasonal and climatic conditions, resources located outside of the school, and other factors. Programs that offer an Associate of Applied Science Degree must include all of the required Career Certificate courses, Technical Certificate courses **AND** a minimum of 15 semester hours of General Education Core Courses. The courses in the General Education Core may be spaced out over the entire length of the program so that students complete some academic and Career Technical courses each semester. Each community college specifies the actual courses that are required to meet the General Education Core Requirements for the Associate of Applied Science Degree at their college.

- In order to provide flexibility within the districts, individual courses within a framework may be customized by doing the following:
  - Adding new student learning outcomes to complement the existing competencies and suggested objectives in the program framework
  - Revising or extending the student learning outcomes
  - Adjusting the semester credit hours of a course to be up 1 hour or down 1 hour (after informing the Mississippi Community College Board [MCCB] of the change)

Appendix C: Recommended Resources/Textbook List

Coding CIP: -		
Book Title	**Author (s)**	**ISBN**
Practical Programming: An Introduction to Computer Science Using Python 3 (2nd Edition)	Gries, Paul, Campbell, Jennifer, Montojo, Jason	
Two Scoops of Django	Daniel Roy Greenfield and Audrey Roy Greenfield	
Head First Java	Kathy Sierra & Bert Bates	
HTML & CSS: Design and Build Websites	Duckett, Jon	
JavaScript and jQuery: Interactive Front-End Web Development	Duckett, Jon	

- For Java, we have a heavy emphasis on an introduction to the Java IDE (Integrated Development Environment). We use IntelliJ IDEA (https://www.jetbrains.com/idea/) as our IDE, and it is worth having an introduction to this for the students because the workflow for writing a Java application can be quite different from the other languages we use.
- For JavaScript, programs can use JQuery
- Might also be interested in:
  - Vanilla JS for DOM manipulation.
  - Fetch for AJAX (asynchronous) requests.
- To prepare for the transition to Django, might have a higher emphasis on templating during our Front End Web section. Use a JavaScript library called Handlebars (https://handlebarsjs.com/) to work on creating the templates.

Might also emphasize using a design library (like Bootstrap https://getbootstrap.com/) to make responsive design easier. This helps the website format better for large (computer) and small (phone) screens.

DJANGO DOCUMENTATION. DJANGO SOFTWARE FOUNDATION, 2017, HTTP://DOCS.DJANGOPROJECT.COM/EN/1.10/. ACCESSED 01 FEB. 2017.