CMPS 300
PROGRAM LANGUAGES
SPRING 2020

2017 - 2020 Catalog Data: CMPS 300 **PROGRAM LANGUAGES**
(Credit, 3 hour) (Lecture, 3 hours). This course is intended to survey the significant features of existing programming languages with emphasis on underlying concepts abstracted from these languages. The structure of simple statements, the structure of algorithmic languages, list processing and string manipulation languages and including procedural, object-oriented, logic programming, and functional languages. Prerequisites: CMPS 201

I. COURSE INFORMATION

Course Information:

CRN	24015
Rubric No	CMPS 300
Section	1
Title	Program Languages
Credit Hours	3

Lectures: 3:30 PM – 4:50 PM, TR
Henry Thurman Jr. Hall,
Lecture Classroom 204

II. TEXTBOOK AND MATERIALS

Main Textbook: Robert W. Sebesta, *Concepts of Programming Languages,* 11th Edition, Addison Wesley Inc., 2015.
ISBN-13: 978-0133943023 / ISBN-10: 013394302X

Reference Textbooks Michael Scott, *Programming Language Pragmatics*, 4th Edition, Elsevier Inc., 2016.
ISBN-13: 978-0124104099

III. INSTRUCTOR INFORMATION

Instructor(s):

Name:	Yaser Banadaki, Ph.D.
Office Location:	E 114 Henry Thurman Jr. Hall, Southern University Baton Rouge
Office Phone:	225-771-3941
Office Hours:	MWF 10:00 AM – 11:00 AM or by appointment
E-mail:	yaser_banadaki@subr.edu

Course Coordinator: Dr. Yaser Banadaki

IV. COURSE LEARNING OUTCOMES

ABET Learning Outcomes:

Each graduate by the time of graduation will demonstrate:

1. An ability to analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solution (ABET Outcome-1),
2. An ability to recognize professional responsibilities and make informed judgements in computing practice based on legal and ethical principles (ABET Outcome-4),
3. An ability to apply computer science theory and software development fundamentals to produce computing-based solution (ABET Outcome-6).

Course Objectives:

The objectives of this course are to cover:

1. the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.
2. the notations to describe syntax and semantics of programming languages.
3. the behavior of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.

Course Learning Outcomes:

Upon completion of this course, students will be able to:

1. demonstrate the ability to identify the issues involved in programming language design and implementation.
2. demonstrate the ability to explain functional, logic, and object-oriented programming paradigms.
3. demonstrate the ability to implement several programming languages
4. demonstrate the ability to identify the issues involved with variable allocation and binding, control flow, types, subroutines, parameter passing.

Course Educational Strategies:

1. Provide clear lectures and discussions of appropriate programming language concepts.
2. Provide students with the opportunity to learn course material through reading and homework assignments.
3. Allow students to demonstrate mastery of the course concepts through submitted exercises such as exams, quizzes, and homework problems.
4. Provide students with the opportunity to learn more about various types of programming languages through their applications to real world problems.

V. COURSE OUTLINE

Course Topics:

This course will cover the following topics (with tentative time throughout the semester):

1.	Intro to Programming Languages	2 classes
2.	History of Major Programming Languages	3 classes
3.	Formal Descriptions of a Programming Language	3 classes
4.	Language Translation Issues	3 classes
5.	Names, Bindings, Type Checking and Scopes	2 classes
6.	Data Types	4 classes
7.	Expressions and Assignment Statements	3 classes
8.	Control Structures	3 classes
9.	Subprograms	3 classes
10.	Implementing Subprograms	3 classes

Course Weekly Content:

Week Topic

1. Preliminaries:

Reasons for Studying Concepts of Programming Languages, Programming Domains, Language Evaluation Criteria, Influences on Language Design, Language Categories, Language Design Trade-Offs, Implementation Methods, Programming Environments.

2. Evolution of the Major Programming Languages:

Zuse's Plankalkül, Minimal Hardware Programming, Functional Programming, The First Step Toward Sophistication, Computerizing Business Records, The Beginnings of Timesharing, Everything for Everybody, Two Early Dynamic Languages, The Beginnings of Data Abstraction, Orthogonal Design, Programming Based on Logic, History's Largest Design Effort, Object-Oriented Programming, Combining Imperative and Object-Oriented Features, Programming the World Wide Web.

3. **Describing Syntax and Semantics**:
   Introduction, The General Problem of Describing Syntax, Formal Methods of Describing Syntax. Attribute Grammars, Describing the Meanings of Programs: Dynamic Semantics
4. **Lexical and Syntax Analysis:**
   Introduction, Lexical Analysis, The Parsing Problem, Recursive-Descent Parsing, Bottom-Up Parsing.
5. **Names, Bindings, Type Checking, and Scopes:**
   Introduction, Names, Variables, The Concept of Binding, Type Checking, Strong Typing, Type Compatibility, Scope, Scope and Lifetime, Referencing Environments, Named Constants, Variable Initialization.
6. **Data Types:**
   Introduction, Primitive Data Types, Character String Types, User-Defined Ordinal Types, Array Types, Associative Arrays, Record Types, Union Types, Set Types, Pointer Types.
7. **Expressions and Assignment Statements:**
   Introduction, Arithmetic Expressions, Overloaded Operators, Type Conversions, Relational and Boolean Expressions, Short-Circuit Evaluation, Assignment Statements, Mixed-Mode Assignment.
8. **Statement-Level Control Structures:**
   Introduction, Compound Statements, Selection Statements, Iterative Statements, Unconditional Branching, Guarded Commands, Conclusions.
9. **Subprograms:**
   Introduction, Fundamentals of Subprograms, Design Issues for Subprograms, Local Referencing Environments, Parameter-Passing Methods, Parameters That Are Subprogram Names, Overloaded Subprograms, Generic Subprograms, Separate and Independent Compilation, Design Issues for Functions, Accessing Nonlocal Environments, User-Defined Overloaded Operators, Coroutines.
10. **Implementing Subprograms:**
    The General Semantics of Calls and Returns, Implementing FORTRAN 77 Subprograms, Implementing Subprograms in ALGOL-Like Languages, Implementing Dynamic Scoping, Implementing Parameters That Are Subprogram Names.

VI. EVALUATION AND GRADING

Grading Distribution

A student's grade at the end of the semester will be determined by following percentages:

Class participation/Activities	10%
Assignments/Quizzes/Projects	30%
Midterm Exam	30%
Final Exam	30%

Students will demonstrate knowledge of the subject through 3 tests, 5 assignments, and the final exam.

Grading Scale

Course grades at the end of the semester will be given based upon performance using the standard grading scale:

90– 100%	A
80 – 89%	B
70 – 79%	C
60 – 69%	D
Below 60%	F

Grades may be curved at the end of session at the instructor's discretion.

VII. COURSE RULES AND PROCEDURES

Exam policy: Exams will emphasize concepts developed in the course. Exams will be closed book and notes unless stated otherwise (if formulas, tables, etc. are needed, the instructor will supply them). Students will not be able to leave the classroom while they are taking an exam. There will be no sharing of any materials (including calculators) during exams. Communicating to one another is not permitted while taking an exam. If there are questions, they should be directed to the instructor. Students who communicate to one another will receive a zero for that exam.

Attendance policy and class participation: Students are expected to be present and on time and participate in class from the beginning throughout the end of the class period. Though attendance is strongly encouraged, merely attending classes does not guarantee a passing grade. It is every student responsibility to make sure that he/she signs the roll in class when one is provided, or he/she pays attention during the roll call.

Makeup test or exam: No makeup test or exam will be given except in the case of emergency such as the student being sick, or he/she is unable to come to class due to some unforeseen event. An official proof MUST be presented to the instructor and student is required to take the makeup test/exam as soon he/she returns to class in the following class session. Failure to comply will result in the grade of zero (0) for the test/exam.

Use of electronic devices while in class: Students are not allowed to use the classroom computers or laptop during the lectures unless authorized by the instructor of this course for the purpose of the course. Please turn off (or place on silence) your cellular phones before the lecture starts.

Missing or late assignments/quizzes/exams: At the instructor's discretion, students may be given opportunity for late submission of an assignment or retake of a quiz or exam upon presentation of a valid excuse.

Academic honesty and plagiarism: Please review the Southern University – Baton Rouge Student Handbook for information regarding the university's academic conduct policy and what constitutes plagiarism. Academic dishonesty and plagiarism will NOT be tolerated.

Assignment policy: Students are NOT allowed to share their assignments or to communicate during the tests or exam. No late assignment will be accepted and no make up for assignments and quizzes.

Change to syllabus: Any aspects of this syllabus may be subject to change. However, any substantive changes affecting distribution of grades for various components of the course will be accompanied with prior notice given to students via class announcement and Blackboard announcement or email.

Livetext Access: Each student is required to have access to LiveText. Southern University and A&M College-Baton Rouge has entered into partnership with LiveText, Inc. to provide online academic resources for student collaboration and learning outcomes assessment. Therefore, all students enrolled in this course are required to purchase a subscription from LiveText, Inc. through the Southern University Bookstore. LiveText, Inc. provides students with the electronic tools and services needed to serve them in their courses and in their career or academic pursuits beyond graduation.

Moodle Access: Southern University and A&M College at Baton Rouge will used Moodle extensively in this course. Moodle is a learning management system designed to help teachers and students communicate effectively online. The course syllabus, class materials (e.g., handouts, PowerPoint slides, journal articles, assignments, readings, etc.) will be placed on Moodle. The student should check Moodle DAILY for all assignments submitted via Moodle. All course communications will be primarily via SUBR email or Moodle. Students are responsible for regularly checking their emails and Moodle. If the student has problems with his Moodle account, he/she should contact Ms. Chrisena Williams-Brown in the Division of Information Technology via email at chrisena_williams@subr.edu or via phone at (225) 771-5017.

Academic Dishonesty: The University defines academic dishonesty as premeditated and un-premeditated fraudulent behavior. Premeditated fraud is defined as conscious, pre-planned, deliberate cheating with materials prepared in advance. Unpremeditated fraud is defined as cheating without the benefit of materials prepared in advance. See the Southern University and A & M College Catalog for a more detailed definition of academic dishonesty. In addition, administrative regulations governing the conduct of students enrolled at the University are contained in the Code of Student Conduct. A copy of the Code of Student Conduct may be obtained from the Office for Student Affairs.

ADA Compliance: Students with documented disabilities who believe that they may need accommodations in this class are encouraged to contact the Disability Services Coordinator in the Office of Disability services, 234 A.C. Blanks Hall, 225-771-3950 (Voice/TTD), 225-771-5652 (Fax), as soon as possible to ensure that such accommodations are implemented in a timely fashion. Students who need accommodations must be registered with the Office of Disability Services. Students are responsible for informing the instructor of any instructional accommodations and/or special learning needs at the beginning of the semester. All discussions will remain confidential.

VIII. COURSE ASSESSMENT

Course Objectives	Course Learning Outcomes	Methods of Assessment	Target	Relationship to Program Learning Outcomes
Objective 1: Students will be able to classify the major programming paradigms, and the principles and techniques involved in design and implementation of modern programming languages.	Students will demonstrate the ability to classify the major programming paradigms including the principles and techniques in modern programming languages.	Outcome Rubrics 1, 4 and 6 will be assessed through 1 tests, 6 assignments, 8 in-class quizzes and the final exam.	70% will perform at the level of performance 2, 3 or 4 in achieving Outcomes 1, 4 and 6.	1, 4 and 6
Objective 2: Students will be able to describe the behavior of simple programs in imperative languages using concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.	Students will demonstrate the ability to provide the description on simple programs in imperative languages using the concepts such as binding, scope, control structures, subprograms and parameter passing mechanisms.	Outcome Rubrics 1, 4 and 6 will be assessed through 1 tests, 6 assignments, 8 in-class quizzes and the final exam.	70% will perform at the level of performance 2, 3 or 4 in achieving Outcomes 1, 4 and 6.	1, 4 and 6