COURSE OUTLINE
Java Programming

Course Description
IN 211. Java Programming. 3 hours credit. This course will enable the student to design, code, and implement software projects in Java. The student will solve coding problems involving control structures, data types, classes, methods, and events. The student will implement inheritance, exceptions, user interfaces, recursion, searching and sorting within a Java program. The student will apply a documented project development cycle to the production of Java applications.

Required Materials
For complete material(s) information, refer to https://bookstore.butlercc.edu

USB digital drive 4GB minimum or equivalent online file storage space. The same drive or storage space can be used in several information technology courses.

Butler-assessed Outcomes
The intention is for the student to be able to
1. Design, implement, compile, debug, test, and run software projects in Java.
2. Design and implement programs using Java in a documented product development cycle.

Learning PACT Skills that will be developed and documented in this course
Through involvement in this course, the student will develop ability in the following PACT skill area(s):
Technology skills
• Discipline-specific technology – Through the creation of applications, the student will demonstrate skills in the field of Java programming.

Major Summative Assessment Task(s)
These learning outcomes and the Learning PACT skills will be demonstrated by
1. Designing and implementing an application using Java in a documented product development cycle.

Skills or Competencies
These actions are essential to achieve the course outcomes:
1. Articulate the relationships between computer systems and programming languages.
2. Describe the basics of Java programming.
3. Write and debug Java code that references objects.
4. Implement control structures and selection expressions within Java code.
5. Use control structures and repetition methods within Java code.
6. Articulate the relationship between Graphical User Interface (GUI) and Object-Oriented Design (OOD).
7. Describe and implement user-defined methods, including void methods and variables as parameters.
8. Describe and implement user-defined classes within Java code, including constructors and finalizers.
9. Discuss and implement arrays within Java code.
10. Articulate the application of inheritance and polymorphism within Java code.
11. Discuss and implement exceptions and events.
12. Implement advanced GUI and graphic design within a Java project.
13. Describe and implement recursion in a Java program.
14. Implement search and sort processing within a Java project.

Learning Units
I. Computers and programming languages overview
   A. Elements of a computer system
   B. Language of a computer
   C. Evolution of programming languages
   D. Internet, World Wide Web, browsers, and Java
   E. Problem analysis, coding, and execution development cycle
   F. Programming methodologies: Structured programming vs object-oriented programming

II. Java fundamentals
   A. Basics of a Java program
   B. Data types
   C. Arithmetic operators and operator precedence
   D. Expressions
   E. Type conversion (casting)

III. Objects
   A. Input/output
   B. Debugging

IV. Control structures and selection
   A. Relational operators
   B. Primitive data types
   C. Boolean operators and logical expressions

V. Control structures and repetition
   A. Looping structures
   B. Debugging loops
   C. Nested control structures

VI. GUI and OOD
   A. GUI components
   B. Object-oriented design
C. Classes and operations

VII. Methods
   A. Predefined methods
   B. User-defined methods
   C. Void methods
   D. Variables as parameters

VIII. User-defined classes
   A. Classes and objects
   B. Classes and methods
   C. Constructors
   D. Finalizers

IX. Arrays
   A. Object arrays
   B. Two-dimensional arrays
   C. Multidimensional arrays

X. Inheritance and polymorphism
   A. Inheritance
   B. Java stream classes
   C. Polymorphism
   D. Abstract methods and classes

XI. Exceptions and events
   A. Java exception hierarchy
   B. Checked and unchecked exceptions
   C. Exception-handling methods
   D. Event handling

XII. Advanced GUIs and graphics
   A. Applets
   B. GUI components
   C. Layout managers
   D. Menus
   E. Mouse and key events

XIII. Recursion
   A. Direct and indirect
   B. Infinite

XIV. Searching and sorting
   A. List processing
   B. Selection sort
   C. Insertion sort
**Learning Activities**
Learning activities will be assigned to assist the student to achieve the intended learning outcome(s) through lecture, instructor-led class discussion, guest speakers, group activities, drills/skill practice, and other activities at the discretion of the instructor.

**Grade Determination**
The student will be graded on learning activities and assessment tasks. Grade determinants may include the following: daily work, quizzes, chapter or unit tests, comprehensive examinations, projects, presentations, class participation, and other methods of evaluation at the discretion of the instructor.