



COURSE OUTLINE

Revised Spring 2008

Course Number COS 102	Course Title Computer Science 1 Algorithms and Programming	Credits 4
Hours: 3 lecture/2 Lab	Pre-requisite: COS 101 or equivalent Co-requisite: MAT 141 or higher-level MAT course	Implementation: Spring, 2008

Catalog description (2006-2009 Catalog):

Algorithm design and object oriented programming in the Java programming language. Topics include data representation, input/output, control structures, exception handling, classes, methods, inheritance, polymorphism, encapsulation, overloading and dynamic memory.

Required texts/other materials: "Java: An Introduction to Problem Solving and Programming", Latest Edition By Walter Savitch, Prentice Hall, Paperback, ISBN: 0131492020. Thumb or Flash Drive.

Revision date: 2/2008

Course coordinator: Don Reichman, 609-570-3857, reichmad@mccc.edu

Information resources: Textbook (described above), JGrasp Integrated Development Environment, Java Development Kit (Sun Systems), instructor's website

Other learning resources: Angel Learning Systems

Course Competencies/Goals (CC/G)

The student will be able to:

1. Be able to design algorithms and then develop solutions using a formal programming language.
2. Have refreshed basic programming skills including sequence, selection, iteration and functions.
3. Have an understanding of the fundamental object oriented programming concepts.
4. Comprehend OOP concepts of inheritance, polymorphism, encapsulation and overloading.
5. Understand and implement basic data types, classes and methods.
6. Understand and be able to write programs that work with streams, files and input/output.
7. Implement simple examples of arrays, strings, vectors and linked lists.
8. Describe event driven programming and the concept of graphical interfaces and applets.

Course-specific General Education Knowledge Goals and Core Skills

General Education Knowledge Goals (GEKG)

Goal 4. Technology. Students will use computer systems or other appropriate forms of technology to achieve educational and personal goals.

MCCC Core Skills (CS)

Goal B. Critical Thinking and Problem-solving. Students will use critical thinking and problem solving skills in analyzing information.

Goal D. Information Literacy. Students will recognize when information is needed and have the knowledge and skills to locate, evaluate, and effectively use information for college level work.

Goal E. Computer Literacy. Students will use computers to access, analyze or present information, solve problems, and communicate with others.

Units of study in detail

Unit I Introduction to Java

Learning Objectives

The student will be able to...

- Explain the history of Java (CC/G 2)
- State the importance of portability (CC/G 3)
- Differentiate between languages and libraries (CC/G 3)
- Understand programming languages and coding styles (CC/G 2)
- Name the "editions" of Java and their differences
- Explain the fundamentals of OOP concepts (CC/G 3)
- Write their first "hello world" Java programs (CC/G 1, GEKG 4, CS B)

Unit II Primitive types, Strings, Simple I/O

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Understand literals and primitive types (CC/G 1, 5)
- Write assignment statements and declare and initialize variables (CC/G 1, 5)
- Explain arithmetic operators and the precedence rules (CC/G 5)
- Calculate accuracy, floating point inaccuracies, conversions and detect overflow (CC/G 5)
- Understand object wrappers for the primitive types and autoboxing and unboxing (CC/G 5)
- Explain the parent class java.lang.Object and its methods (CC/G 5)
- Work with the java.lang.String class and basic string class operations (CC/G 7)

Unit III Flow of Control

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Describe expression statements versus selection statements versus looping statements (CC/G 2)
- Evaluate Boolean expressions, standard and short circuit evaluation, forcing complete evaluation (CC/G 2)

- Code the if-else statement, nested if-else statements and switch statements (CC/G 2)
- Code the while, do-while and for statements and use the comma operator (CC/G 2)
- Understand counters versus sentinels and use break and continue statements, pros and cons (CC/G 2)
- Debug looping statements, find boundary conditions and trace variables (CC/G 2)
- Use the ternary operator (CC/G 3)

Unit IV Classes and Methods

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Write class and method definitions, using separate .java files and separate compilation (CC/G 5)
- Invoke methods, use return values, void methods and return statements (CC/G 5)
- Understand the “this” object (CC/G 5)
- Use local variables, blocks and scope (CC/G 5)
- Utilize information hiding and encapsulation (CC/G 5)
- Explain access modifiers and accessor methods versus mutator methods (CC/G 5)
- Describe objects and reference, memory addresses, reference types and the new operator (CC/G 5)
- Differentiate between call by value versus call by reference (CC/G 5)
- Write methods that call other methods (CC/G 5)
- Differentiate between static methods and static variables (CC/G 5)
- Understand method overloading (CC/G 5)
- Write constructors (CC/G 5)
- Use static, final and enumerated types (CC/G 5)

Unit V Arrays

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Describe arrays as objects
- Create, initialize and access arrays
- Understand array indices, stepping through with loop statements, and the length instance variable
- Pass arrays as method arguments and use methods that return arrays
- Work with partially filled arrays
- Search through and sort the elements of an array
- Work with multidimensional arrays and ragged arrays

Unit VI Inheritance and Polymorphism

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Understand inheritance, parent/child relationships, base/derived classes
- Use the extends keyword
- Override a method and describe overriding versus overloading
- Use the final modifier
- Write constructors in derived classes and the super() method
- Call an overridden method
- Differentiate between the “is-a” and “has-a” relationships
- Describe polymorphism, dynamic binding

-

Unit VII Exception Handling

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Describe exceptions in Java, try-catch blocks, throw statements
- View exceptions as objects
- Use the getMessage() method

- Work with predefined exception classes
- Define their own exception classes
- Code multiple catches and/or throws per try block and nested try-catch blocks
- Describe run-time internals, the heap, memory leaks, and garbage collection
- Use the assert statement

Unit VIII Streams and I/O

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Understand concept of streams, Java I/O design philosophy, and the java.io package
- Work with keyboard I/O, text file I/O and binary file I/O
- Work with PrintWriter and BufferedReader
- Work with DataOutputStream and DataInputStream
- Work with IOException and EOFException
- Work with FileInputStream and FileOutputStream
- Describe the File class

Unit IX Dynamic Data Structures

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Understand and write programs involving vectors
- Understand and write programs involving linked lists

Unit X Advanced Topics

Learning Objectives (All use GEKG 4, CS Goal B)

The student will be able to...

- Understand and write programs involving recursion such as the binary search
- Describe GUI basics, the concept of event driven programming, and using Java Swing
- Understand and write Java applets
- Understand and write programs involving threads and multithreaded applications
- Understand and write programs involving Java network client/server programming

Evaluation of student learning: Specific methods for evaluating student progress through the course is up to the discretion of the instructor. In general, exams are intended to assess a student's knowledge of the learning objectives described above. Labs and homework assess a student's ability to solve problems using a formal programming language(CC/G 1). Below are two examples used in past COS-102 courses balancing lab, lecture, and home work.

Example 1:

Midterm Exam = 30% of the grade

Ten Labs = 30% of the grade

Homework = 10% of the grade

Final Exam = 30% of the grade

Example 2:

Three 1-Hour Exams = 45% of the grade

Ten Labs = 30% of the grade

Final Exam = 25% of the grade

Academic Integrity Statement: Mercer County Community College is committed to Academic Integrity—the honest, fair and continuing pursuit of knowledge, free from fraud or deception. This implies that students are expected to be responsible for their own work and that faculty and academic support services staff members will take reasonable precautions to prevent the opportunity for academic dishonesty.

See http://www.mccc.edu/admissions_policies_integrity.shtml for a complete explanation of policies and procedures regarding academic integrity and academic integrity violations.

