231.348.2165 :: hbates@northmen.org

CIS 225

Introduction to Java Programming

Description

CIS 225 Introduces students to the Java programming language. Object oriented concepts & methodologies are taught and practiced. The course focuses directly on UML, classes, objects, fields, constructors, methods, enums, Java control structures + logic, operators, coding style, Java library classes, collections, GUI programming, user and file I/O, and program documentation including Javadoc. This course prepares the student for the Oracle Java SE 8 Programmer I exam. Additionally, an introduction to Android development is explored at the end of the course.

Logistics

Class location:	Petoskey Middle School Media Center Computer Lab
Class time:	Mondays, 6:00pm - 10:00pm
Instructor:	Howard Bates (Please call me Howard)
Office:	Petoskey High School
Office hours:	By arrangement (typically 7:00am - 4:00pm)
Contact info:	hbates@ncmich.edu -or- 231.348.2165
Text:	Objects First with Java, 8th edition. Barnes, Kolling 2012
Website:	http://www.petoskeyschools.org/ncmc/225
Prerequisite:	CIS 104
Required:	JDK 8, IntelliJ IDE, BlueJ Editor, GitHub account
Helpful:	IntelliJ IDEA Keymap,

Policies

Course Structure:
The class is structured to be 1.5 hours of instruction/discussion followed by 1.5 hours of lab time each week. You should expect to spend additional time outside of class completing your assignments.

Student Service Information:
Information is HERE. Academic calendar is HERE. FERPA is HERE.

Disability Information:
Learning Support Services: Reasonable accommodations can be provided for students with documented disabilities. Please contact Learning Support Services at: (231) 348-6682 or lss@ncmich.edu, located in room 533 in the SCRC.

Course Policies:

Attendance
It is your responsibility to attend all class sessions. If you can't make it, you need to notify me via email. I do not need a reason, just notification. If you fail to notify me more than once, I will lower your grade 20%. If you fail to notify me after that, I will ask you to drop the class for no credit.

Lateness
As long as you are not disruptive, I do not care how late you are.

NCMC Snow Cancellation Policy
If the public schools cancel class in Cheboygan or Gaylord, NCMC classes will also be cancelled. If Petoskey schools are cancelled, NCMC may be open or may just delay opening until later that day. It is best to sign up for My Alerts, found on the portal.

Class Participation
Engage in our discussions as you see fit.

Academic Dishonesty
Academic Dishonesty, misconduct, cheating or plagiarism or other forms of academic dishonesty including acquisition without permission of tests or other academic materials. Included are those students who aid and abet, as well as those who attempt such behavior. Plagiarism includes, but is not limited to, the use whether by paraphrase or direct quotation, of published or unpublished work of another person without full and clear attribution. It also includes the unacknowledged use of materials prepared by another person or agency engaged in the selling of term papers or other academic materials. Incidents of academic dishonesty shall be dealt with according to the procedures outlined in Academic Dishonesty as found in the student handbook. Any act of academic dishonesty in this course will result in a zero (0) for the assignment and possible failure of the course.

Late Assignments
All assignments must be turned in on the due date. If you can't accomplish this, I must be notified by email. I will then decide if I will accept the late assignment for full credit. The late penalty is as follows: First assignment one week late = 25% markdown. Any more than this will not be accepted for credit.

Test Make up
You may make up a missed test the following week or by appointment for full credit.

Electronic Device usage in class

As long as you are not disruptive, I don't care what you choose to use.
(phone, .mp3 player, laptop, etc.)

Objectives

Programming Ten Commandments
[] Self-documenting code
[] Consistent, proper style
[] Prefer numeric over String
[] Explicit over implicit
[] NO magic numbers
[] Prefer local over global
[] Validate, distill, & sanitize input
[] Limit recursion
[] Loose coupling & high cohesion
[] Practice algorithms

Modern Programming
[] Integrated development environments
[] Modern programming paradigms (Agile, Pair, XP, aspect)
[] Unified modeling language
[] Project management/Customer Service
[] Revision control systems (Git)
[] Code debugging & Unit Testing

Java Basics
[] Define the scope of variables
[] Define the structure of a Java class
[] Create executable Java applications with a main method
[] Import other Java packages to make them accessible in your code
[] Use the Java API

Working With Java Data Types
[] Declare and initialize variables
[] Differentiate between object reference variables and primitive variables
[] Read or write to object fields
[] Explain an object's lifecycle
[] Call methods on objects
[] Manipulate data using the StringBuilder class and its methods
[] Create and manipulate strings
[] Use Java random functionality
[] Use anonymous objects
[] Enumerated types
[] File input/output

Using Operators and Decision Constructs
[] Use Java operators
[] Use parentheses to override operator precedence
[] Test equality between strings and other objects using == and equals()
[] Create if and if/else constructs
[] Use a switch statement

Creating and Using Collections
[] Declare, instantiate, initialize and use a one-dimensional fixed-size array
[] Declare, instantiate, initialize and use multi-dimensional fixed-size array
[] Declare and use an ArrayList
[] Declare and use an Iterator
[] Declare and use a HashMap
[] Declare and use a HashSet

Using Loop Constructs
[] Create and use while loops
[] Create and use for loops including the enhanced for loop
[] Compare loop constructs
[] Use break and continue

Working with Methods & Encapsulation
[] Create methods with arguments and return values
[] Apply the static keyword to methods and fields
[] Create an overloaded method
[] Differentiate between default and user-defined constructors
[] Create and overload constructors
[] Apply access modifiers
[] Apply encapsulation principles to a class
[] Determine the effect upon object references and primitive values when they are passed into methods that change the values

Working with Inheritance
[] Implement inheritance
[] Develop code that demonstrates the use of polymorphism
[] Differentiate between the type of a reference and the type of an object
[] Determine when casting is necessary
[] Use super and this to access objects and constructors
[] Use abstract classes
[] Use interfaces

Handling Exceptions
[] Differentiate among checked exceptions, RuntimeExceptions and Errors
[] Create a try-catch block and determine how exceptions alter normal program flow
[] Describe what exceptions are used for in Java
[] Invoke a method that throws an exception
[] Recognize common exception classes and categories

Introduction to Android Development
[] Android SDK
[] Google Play Development
[] Device Testing
[] UX Design

Grading

RUBRIC

Letter Grade	Points	Percent (%)	Honor Points
A	930-1000	93-100	4.00
A-	900-929	90-92.99	3.67
B+	880-899	88-89.99	3.33
B	830-879	83-87.99	3.00
B-	800-829	80-82.99	2.67
C+	780-799	78-79.99	2.33
C	730-779	73-77.99	2.00
C-	700-729	70-72.99	1.67
D+	680-699	68-69.99	1.33
D	630-679	63-67.99	1.00
D-	600-629	60-62.99	0.67
E	000-599	0-59.99	0.00