Object Oriented Programming (Java)
Course Syllabus

Instructor: Mr. Tom Meyer
E-mail: tmeyer@imsa.edu
Phone: 630-907-5482

Office hours: Drop in the CS office and see if we are available. All three CS teachers can help you with OOP questions.
Mr. Meyer: 1:15 pm – 2:15 pm or by appointment.

Attendance: We strictly follow IMSA attendance policy; NO EXCEPTIONS.

Materials used:

1. Text: *Objects First With JAVA: A Practical Introduction Using BLUEJ 6th Ed.*, Barnes, D.J. and Kölling, M.

2. Online resources

Course Description:

This one-semester course is designed to teach the fundamental concepts of computer programming. The course emphasis is on the creation and use of "objects" as the basic tool for developing various program algorithms (such as finding the lowest common divisor, sorting an array, finding prime numbers, etc.), data structures (such as arrays, structures, strings, maps), and programming processes (such as manipulating data files, passing parameters by value and by reference). Throughout the course there is an emphasis on the use of existing "classes" and the development of new, project-related classes.

Course components:

Exercises: Must be completed on a timely basis. Will be checked regularly during class/lab time.

Projects: Several programming projects will be assigned throughout the semester. Ample time is allowed for each project in and out of class time. Final project will be a design project which replaces the final exam. Some GUI components are included in the final project.

Labs: Labs are designed to be completed in class. As such the labs are due by the end of the class the day they are assigned. The labs are meant to help students exercise specific concepts covered in (or out of) the book.

Quizzes/Exams: Pencil and paper tests/quizzes will be given in addition to programming projects. All the tests/quizzes are comprehensive and will include all the material taught in class as well as some material that students are responsible for learning from their projects/assignments and/or online resources.

Late Policy: Late homework/projects will be severely penalized:
by end of the day: -20%
by beginning of the next class: -50%

All the assignments are to be submitted by the beginning of the class on a due date.

Academic Honesty: All programs must be your own work. Copies of another's work will be considered plagiarism and treated accordingly. IMSA's plagiarism policy will be strictly enforced.

Class Rules: No food or drinks will be allowed in the class. No gaming in the class (unless someone in class wrote it using Java). No head phones allowed during class. Every student will stay on task during the class time and will not leave the class till dismissed or he/she risks being marked absent.

Quarterly grades will be averaged using the following weighting:

Projects	30%
Quizzes/Tests	30%
Exercises	15%
Labs	20%
Participation/Organization	5%

Semester grades will be averaged using the following weighting:

Cumulative semester work	95%
Semester Final Project or Final Exam	5%

The course syllabus contents are subject to change.

The grades are computed to two decimal places. No rounding will be done so as to change the letter grade.

BLUEJ Website: http://www.bluej.org/

Oracle Java Tutorials: http://docs.oracle.com/javase/tutorial/

Moodle website: https://courses.imsa.edu

Java API: https://docs.oracle.com/javase/8/docs/api/index.html

Topics to be covered from text book in addition to other topics from online resources:

1.	Introduction to OOP: Objects and Classes objects, classes, methods, parameters, return values, object’s state, data type and object interaction.	Chapter 1
2.	Understanding class definition Class header, attributes, constructors, assignments, accessors and mutators methods (encapsulation), class and local scope. Reflecting on design.	Chapter 2
3.	Object Interaction OO design concepts: encapsulation, abstraction, modularization, UML class diagram and UML object Diagram, primitive vs object type, constructor overloading. Code debugging using BlueJ built-in debugger.	Chapter 3
4.	Grouping Objects Collections: lists, generic classes and iterators. Primitive arrays and the ArrayList class (array of objects). Loops Well behaved objects	Chapter 4, Chapter 7
5.	More Sophisticated behavior Javadoc, Random class, packaging, access modifiers(public, private), constants, the String class. More Data structures: HashMap and HashSet classes	Chapter 5
6.	Designing class more on UML class diagram, coupling cohesion, code duplication, design guidelines and the main() method using command line compiling	Chapter 6
7.	Introduction to Inheritance hierarchies and rules of inheritance. Advantages and design	Chapter 8 ((intro only)
8.	Introduction to GUI Swing, AWT basic form	Chapter 11 (intro only)