Comprehensive Course Syllabus

Advanced Programming

Course Description:
This course is designed to advance the student's understanding of computer science with two primary objectives: to extend the student's understanding of object oriented programming (OOP) using the java programming language and to examine a variety of data structures. The topics covered may assist the student in his/her preparation for the AP Computer Science A exam.

INSTRUCTOR(S):

- Name(s): Dr. Phadmakar Patankar
- Office Number(s) (When and where you are available for help.): A156

  All Days: 9:15am – 9:45am or by Appointment
- Telephone number(s): 630-907-5479
- Email address: ppatankar@imsa.edu

Meeting Days, Time and Room(s)
All Class Days: 10:00am – 10:55am
All class Days: 11:00am – 11:55pm

Text(s) / Materials:
Text: Big Java by Cay Horstmann

Use google classroom for assignments and dissemination of information for this class.

References:

AP Central:
http://apcentral.collegeboard.com/apc/public/courses/teachers_corner/4483.html
Sun JAVA Website: http://www.oracle.com/technetwork/java/index.html
Java API: http://docs.oracle.com/javase/7/docs/api/
Eclipse: http://www.eclispe.org

Essential Content:

We have used, “A Model Curriculum for K-12 Computer Science” published by the Computer Science Teachers Association (CSTA) to refer to the standards use for this course for course.

Since IMSA has not adopted Computer Science standards yet, we will follow the guidelines of National Educational Technology (NET) standards for students (see **http://www.iste.org/Content/NavigationMenu/NETS/ForStudents/2007Standards/NETS_for_Students_2007.htm**) :

Creativity and Innovation

Students demonstrate creative thinking, construct knowledge, and develop innovative products and processes using technology. Students:

a. apply existing knowledge to generate new ideas, products, or processes.
b. create original works as a means of personal or group expression.
c. use models and simulations to explore complex systems and issues.
d. identify trends and forecast possibilities.

Critical Thinking, Problem Solving, and Decision Making

Students use critical thinking skills to plan and conduct research, manage projects, solve problems, and make informed decisions using appropriate digital tools and resources. Students:

a. identify and define authentic problems and significant questions for investigation.
b. plan and manage activities to develop a solution or complete a project.
c. collect and analyze data to identify solutions and/or make informed decisions.
d. use multiple processes and diverse perspectives to explore alternative solutions.

Digital Citizenship

Students understand human, cultural, and societal issues related to technology and practice legal and ethical behavior. Students:

a. advocate and practice safe, legal, and responsible use of information and technology.
b. exhibit a positive attitude toward using technology that supports collaboration, learning, and productivity.
c. demonstrate personal responsibility for lifelong learning.

d. exhibit leadership for digital citizenship.

Technology Operations and Concepts Students demonstrate a sound understanding of technology concepts, systems, and operations. Students:

a. understand and use technology systems.
b. select and use applications effectively and productively.
c. troubleshoot systems and applications.
d. transfer current knowledge to learning of new technologies.

SSLs and Outcomes:

I. Developing the Tools of Thought IA, IB, IC, ID
II. Thinking about Thinking IIB
III. Extending and Integrating Thought IIIA, IIIB
V. Thinking and Acting with others VA, VB

Student Expectations:
All students are expected to

- be involved in class discussions and explorations, both large and small group.
- attend all the classes and be on time.
- complete all daily assignments, labs, and projects in a timely manner.
- take responsibility for learning certain basic skills and relationships.
- take responsibility for seeking additional help as it is needed.
- collaborate with each other and contribute to each other's learning
- follow the guidelines of Students Handbook about ethical behavior and plagiarism.

Enthusiastically engage in class discussions and explorations. Students are expected to complete the reading assignment before coming to the class and actively participate in classroom discussions/reviews. Students are also expected to take notes of important topics.

Complete all assignments in a timely fashion. Late assignments will be heavily penalized: 20% penalty for turning in the assignment by 4:00pm on the same day it is due, 50% penalty for turning in at the beginning of the next class meeting, **No assignment will be accepted after the second class meeting.**

Attend all scheduled meetings. IMSA attendance policy is strictly enforced. Students will be dropped out of the class if the attendance policy is not met.

Respect lab rules both in and out of class. Students not staying on task in the lab will be

marked absent. No food or drinks will be allowed in the lab. No gaming in the lab (unless someone in class wrote it in Java). No head phones allowed in the lab.

Follow institute guidelines for academic honesty. All programs/exercises must be your own work. Students could be asked to explain their code of any assignment at any time. Copies of another's work will be considered plagiarism and treated accordingly.

Assessment practices, procedures, and processes:

Students will be assessed upon their participation in and contribution to the class, the quality of the programs they submit (both form and function), and the personal growth they achieve which raises their level of understanding in computer science (by demonstrating their knowledge on quizzes. All the quizzes/tests are closed book). They will also be graded on both written and oral presentation skills through at least one presentation/paper. Students may be assessed by their peers on mini quizzes.

Each quarter grade will be averaged using the following weighting:

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

Semester grade will be averaged using the following weighting:

Cumulative semester work	95%
Semester final project or exam	5%

Weights of components of the semester grade:

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.

Sequence of Topics and Activities:

The following chapters (or part of the chapters) from the text book would be covered:
Chapter 7 – Designing Classes
Chapter 9 – Interfaces and Polymorphism
Chapter 11 – Inheritance
Chapter 13 – Array Lists and Arrays
Chapter 14 – Exception Handling

Chapter 17 – Recursion
Chapter 18 – Sorting and Searching
Chapter 19 – An Introduction to Data Structures
Chapter 20 – Advanced Data Structures

In addition, projects may be assigned using Graphical User Interface (GUI).

The syllabus and the classroom policy might change at the discretion of the teacher.