

**Universidad de Puerto Rico
Recinto Universitario de Mayagüez
Departamento de Ingeniería Eléctrica y Computadoras**

**Curso ICOM 4015: Programación Avanzada (Computación I)
Primer Semestre 2008-2009**

Prof. Bienvenido Vélez¹

Oficina: Stefani 701

Teléfono: 832-4040 x5827

Horas de Oficina: Martes y Jueves 1:45-5:00 PM o por acuerdo

E-mail: bvelez@ece.uprm.edu

Web: <http://www.ece.uprm.edu/~bvelez>

Sección 66: MJ 12:30 PM - 1:45 PM en S203 LAB: V 7:30 AM - 9:20 AM en S121

Sección 116: MJ 5:00 PM - 6:15 PM en S227 LAB: J 7:30 AM - 9:20 AM en S121

Introducción

En este curso se estudiarán técnicas avanzadas de programación de computadoras, con énfasis en la programación orientada a objetos y el desarrollo de estructuras de datos. Se asume que el estudiante tiene conocimientos previos en el área de computación, al menos al nivel de lo que se cubre en el curso INGE 3016². Estaremos usando el lenguaje *Java*. En general estudiaremos sobre el uso de subprogramas, estructuras de control, referencias, arreglos, clases, conceptos de estructuras de datos, conceptos importantes de la recursión, su implementación y su aplicación al desarrollo de algoritmos. Veremos sobre herencia, interfaces y "method overloading". Se estudiarán distintos problemas clásicos (por ejemplo, búsqueda y ordenamiento) y soluciones comunes, además de conceptos básicos de estructuras de datos y estructuras de datos básicas.

Objetivos

Los temas y tareas con los que nos envolveremos en este curso le permitirán, entre otras cosas:

1. Conocer en términos generales lo que representa la disciplina de computación con sus áreas fundamentales y familiarizarse con la jerga de ésta.
2. Exponernos al tipo de problema que se estudian en cada una de estas áreas y experimentar con algunos con el desarrollo de programas aplicados a los mismos.
3. Conocer las operaciones fundamentales que puede llevar a cabo la computadora, lo que implicará, implícita o explícitamente, el tipo de instrucción que se puede esperar en un algoritmo a ser ejecutado por la computadora. Esto lo veremos, en la mayoría de los casos mediante el estudio del efecto de cada instrucción del lenguaje Java.
4. Poder entender programas escritos en el lenguaje Java y, con mínimo esfuerzo, con la ayuda de manuales o textos, poder entender otras instrucciones y partes del lenguaje no discutidas en clase: sus reglas gramaticales y su efecto al ser ejecutadas por la computadora (lo que ésta hace cuando las ejecuta).
5. Poder desarrollar algoritmos para resolver problemas de complejidad mediana en los que se requieran estructuras de datos simples y técnicas de programación básicas, y codificar dichos algoritmos usando el lenguaje Java.
6. Conocer sobre aplicaciones comunes en las que la computadora es de gran utilidad y distintas técnicas para trabajar con problemas relacionados: (por ejemplo, ordenamiento y búsqueda de datos).

¹ Prontuario modificado de uno originalmente creado por el Prof. Pedro I. Rivera Vega.

² Es requisito el haber aprobado el curso INGE 3016 para poder estar matriculado en ICOM 4015.

7. Conocer sobre la importancia que tiene la profesión de Ingeniero/a de Computadora en la sociedad moderna y sobre las reglas de comportamiento ético que deben seguirse al ejercer la misma.

Lenguaje de Programación a Estudiar

El lenguaje de programación que estaremos estudiando es el lenguaje Java. El aprender un lenguaje de programación en la clase se hace con el propósito de tener una herramienta que nos permita poner en práctica todos los conceptos a estudiar que son esenciales para el desarrollo de algoritmos e implementación de los mismos en la computadora. Y al aprender ese lenguaje estaremos a la vez aprendiendo muchos de esos conceptos. El lenguaje a estudiar es un lenguaje de uso común en muchas aplicaciones reales hoy día. Es también uno de los lenguajes de mayor uso en cursos como éste y en cursos posteriores en otras universidades. Todo lo que estudiemos en la clase en cuanto al desarrollo de algoritmos y programas lo estaremos viendo en base a este lenguaje de programación. Los conceptos a estudiar en clase son válidos también en la mayoría de los lenguajes de programación que se usan hoy día y se aplican de manera similar en la mayoría de ellos. Aprender a programar en otros lenguajes similares simplemente requerirá, en la mayoría de los casos, de un manual o texto que nos explique el formato de las instrucciones equivalentes en dicho lenguaje.

Evaluación

La evaluación en este curso consiste de exámenes, trabajos de laboratorio y asignaciones para entregar (proyectos de programación). La nota final que obtenga el estudiante estará basada en su desempeño en las distintas tareas de evaluación. En cada tarea el/la estudiante obtiene una nota parcial final. Cada una de estas tareas se refleja en la nota final de acuerdo al peso que se ilustra en la siguiente tabla.

Actividad	Peso en el Promedio Final
Exámenes Parciales (3)	50%
Proyectos de Programación	15%
Laboratorio	10%
Asistencia a Clases ³	05%
Examen Final	20%

Exámenes

Debido a la naturaleza del curso, **todos los exámenes son acumulativos**, requiriendo material cubierto hasta el momento del examen. Pero el énfasis en cada caso se dará al material cubierto desde el examen previo. **Se ofrecerán tres exámenes parciales en días y horarios que serán avisados con al menos dos semanas de anticipación (viernes)**. Cada examen parcial tendrá una duración de no menos de 75 minutos y enfatiza en el material presentado en la tabla que desglosa la secuencia de tópicos a discutir. Este material puede variar, pero en tal caso se les notificará con suficiente tiempo. El **examen final** incluirá preguntas relativas a **todo el material del curso** y se ofrecerá en fecha y horario a decidir por las autoridades pertinentes.

Proyecto de Programación

En este curso se asignarán varios proyectos de programación. En cada caso se le especificará una tarea, la cual, una vez completada, implementará un sistema con la funcionalidad que finalmente se persiga. Para cada proyecto van a recibir un documento que se les entregará con al menos dos semanas de anticipación a la fecha en que se vence el mismo. Ese documento ha de contener las especificaciones del proyecto particular: lo que tiene que hacer, la manera en que debe entregar, lo que tiene que entregar y la fecha de entrega. Es importante recalcar que estos trabajos tienen que entregarse a tiempo, en o antes de la fecha establecida en cada caso. En cada caso se les indicará la penalidad que se estará adjudicando como consecuencia de entregar un trabajo tarde.

³La nota parcial final en este renglón se calcula usando la siguiente expresión: **Max (0, 100-x*20)**, donde **x** es el número de ausencias. El primer día de clases (7 de agosto 2008) no se cuenta.

También es importante que tenga en cuenta que **el trabajo en cada proyecto debe ser individual**, excepto si en las especificaciones del proyecto mismo se establece otra cosa en alguna que otra parte del mismo. **Con respecto a esto es importante que tenga en cuenta que de tener evidencia de que alguien se ha copiado alguna parte del proyecto (o un examen) de alguna fuente (otro estudiante, Internet, etc.), todos los que hayan participado en dicha actividad se corren el riesgo de ser sometidos a sanciones ya establecidas en los reglamentos de la Universidad de Puerto Rico. No se corra el riesgo!!!**

Recuerde, usted está tomando este curso porque quiere aprender y porque lo que aprenda aquí es esencial para poder tener éxito en cursos posteriores del programa ICOM, particularmente en aquellos en que se requiere del desarrollo de software, y en su carrera profesional. La experiencia muestra que **leer y trabajar** en los detalles de los proyectos, laboratorios y ejercicios asignados es la mejor manera de aprender. Usted notará que, si lo hace así, el estudiar para los exámenes se le hará mucho más fácil. **En los exámenes habrá preguntas relativas a las etapas del proyecto, ejercicios de laboratorio y otros ejercicios asignados.**

Usted no debe limitar su uso del centro de cómputos a trabajar en las asignaciones para entregar, puede utilizarlo en cualquier momento que le sea conveniente (y si hay cabida) para ayudar a entender instrucciones y conceptos estudiados en clase. De hecho, esto último es altamente recomendable pues es una de las mejores formas de aprender.

Importante: En los casos en que se entienda necesario, **la corrección de un proyecto puede incluir una entrevista individual con cada estudiante y/o grupo de trabajo.** En dicha entrevista se les pueden hacer preguntas relativas a su solución y a cómo extendería la misma para lograr alguna otra funcionalidad no necesariamente especificada, pero que el/la corrector/a entienda que debe poderse hacer sin mucha dificultad. La puntuación que usted obtenga en cada etapa del proyecto entregado les será notificada en aproximadamente tres semanas a partir de la fecha límite establecida para la entrega de la misma. Se incluirá descripción de las razones por las cuales ha perdido puntos (si ese es el caso). La entrega de estos resultados se hará durante la hora de la clase. Luego de esa fecha en la que se entregan los resultados en la clase usted tendrá la oportunidad de hacer reclamaciones durante las **dos semanas** siguientes⁴. Si usted no está presente al momento en que se entreguen los resultados de un proyecto, es responsabilidad suya el reclamar sus resultados durante horas de oficina. **Recuerde, el periodo de dos semanas para hacer reclamaciones comienza a partir del día en que se entregan los resultados en la clase, independientemente de si usted no ha recibido los mismos por razones de ausencia.**

Laboratorios

El curso incluye una sesión de laboratorio, la cual se ofrece una vez a la semana (vea su programa de clases). **La asistencia al mismo es obligatoria.** En este laboratorio se estarán poniendo en práctica distintos conceptos estudiados en la clase, además de introducir conceptos importantes sobre el uso de las herramientas de desarrollo de programas que nos provee el sistema Linux. En una sesión de laboratorio usualmente se asignan ejercicios relacionados al tema en discusión, los cuales tiene que entregar funcionando correctamente. Los mismos serán corregidos y acumulados para su nota final en esta parte del curso. Recibirán más detalles de esto en el laboratorio mismo. Recuerden que la nota que usted obtenga en esta sesión de laboratorio tendrá un peso de 10% en la nota final que va a obtener en el curso.

Asistencia a Clases

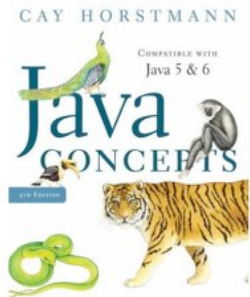
La asistencia a clases, así como al laboratorio (según lo establece su programa de clases) es obligatoria. Note que la misma tiene un peso de un 5% en el cómputo del promedio final. La nota parcial final en este renglón se calcula usando la siguiente expresión: **$\text{Max}(0, 100 - x \cdot 20)$** , donde **x** es el número de ausencias. El primer día de clases (7 de agosto de 2008) no se cuenta. Note que 5 ó más ausencias representará una puntuación de 0 en esta categoría.

⁴Excepto en el último proyecto, pues el mismo se corrige durante el tiempo de exámenes finales, y muy cercano a la fecha límite para la entrega de notas. Además este periodo de tiempo es final y firme e independiente de si usted está presente en la clase al momento de entregar el trabajo corregido. Esta regla también aplica a exámenes corregidos y entregados en clase.

Si usted tiene que ausentarse de alguna clase por razones de peso, tendrá que mostrar evidencia en las horas de oficina. **No se acepta como excusa válida el que usted se ausente a alguna sesión de nuestras clases o del laboratorio por razón de tener que tomar un examen (o alguna actividad de ese tipo) de alguna otra clase en alguna hora que confliga con las asignadas a este curso.**

Libro de Texto

El libro de texto para este curso es el siguiente:

Java Concepts (5th Edition) Cay Horstmann Wiley; 5th edition (March 5, 2007) ISBN 978-0-470-10555-9	
---	--

Website de la Clase

Podrás acceder al website de la clase en:

<http://www.ece.uprm.edu/~bvelez/courses/Fall2008/icom4015/icom4015.htm>

En el mismo estaremos colocando fechas importantes, materiales y enlaces de referencia adicionales durante el transcurso del semestre.

Referencias

En general, cualquier libro de referencia al lenguaje Java o de introducción a la computación le debe ser de utilidad como libro de referencia para este curso. También debe conseguirse un libro sobre el sistema Unix o Linux, ya que ese es el sistema que estaremos usando en el laboratorio. De hecho, libros en esta categoría ya no deben faltarle en su biblioteca privada.

Daremos información sobre libros específicos más adelante en el curso, pero échele un vistazo a las páginas siguientes en internet:

- Página del texto: www.horstmann.com... **busque bajo "Java Concepts"**
Es necesario que se conecte a esta página. En la misma encontrará información muy valiosa sobre el contenido del texto, código de ejemplos que se discuten en el texto, presentación de los tópicos del curso en formato html, cómo usar el sistema Java, etc.
- Página oficial de Java: <http://java.sun.com>
- Un tutorial de Java: www.ibiblio.org/javafaq/javatutorial.html
- [Java™ Platform, Standard Edition 6 API Specification](http://java.sun.com/javase/6/docs/api/)

Si busca usando "search engine" por tópicos como "java tutorial", o algo de ese tipo, va a encontrar muchos otros lugares en el internet en donde hay información de utilidad sobre el lenguaje Java.

Bosquejo de Tópicos a Cubrir

En las páginas siguientes se presentan detalles de los tópicos a cubrir en el curso, día por día. Note que este es un detalle aproximado, pues puede ser que nos adelantemos o nos atrasemos dependiendo de la necesidad en casos particulares. El estimado de tiempo de algunos casos está basado en que ustedes ya deben tener un conocimiento de tópicos similares usando los lenguajes similares a Java (lo que debe haber estudiado en el curso INGE 3016). Según vaya progresando el semestre, puede ser que, por falta de tiempo, se decida pasar por alto alguno que otro tópico entre la lista que se incluye. En ese caso, estaríamos saltando tópicos que no sean esenciales para cursos futuros.

Para cada t3pico estudiado se sugieren ejercicios que se encuentran al final de los cap3tulos correspondientes en el texto. El prop3sito de estos ejercicio es que usted se mantenga practicando sobre los t3picos que se estudian en el curso, los mismos le deben dar una idea de c3mo van a ser los ex3menes. Les aseguro que si usted hace esos ejercicios, con dedicaci3n, no va a tener problema con los ex3menes. No es que en los ex3menes se vayan a preguntar los mismos ejercicios, pero **la experiencia que usted va a ganar resolviendo este tipo de ejercicio asignado es necesaria para salir bien en los ex3menes. Recuerde, es necesario dedicar tiempo a sus cursos, 3ste no es la excepci3n.**

Para cada actividad, debe tomar como asignado para la lectura el material correspondiente **en el texto**. Adem3s puede sugerirse alguna lectura externa o material dado en la clase.

Program of the Course – List of topics by Day ...⁵

Day	Topic or Activity
1	General discussion about the course: the main objectives, the textbook, evaluation criteria, laboratory sessions, contents of the course and of each exam, dates of exams, and general idea of programming projects.
1-2	Chapter 1 : Introduction to computing (most of this material should be a general review for all of you - remember INGE 3016), introduction to the Java system , and how to get (download) it free... visit http://java.sun.com . General format of a Java program, compilation and execution of Java program, and recognition of errors. Extra reading assignment: the material provided in the web page of the author of the textbook (http://www.horstmann.com/), and of the book (http://www.horstmann.com/ccj.html), about how to create, compile, and execute Java programs. Suggested Exercises⁶: R1.1 – R1.12, P1.1 – P 1.8
3	Chapter 2: Using Objects. Introduction to the use of objects and classes in Java. Declaration of variables to reference objects, operating with objects. Methods, parameters, return values, and application of methods to object. API documentation. Suggested Exercises: R2.1 – R2.12, P2.1, P2.2, P2.4, P2.9, P2.10
4-5	Chapter 3: Implementing Classes. How to implement classes in Java, implementation of simple methods, constructors, and instance and local variables. Suggested Exercises: R3.1 – R3.12, P3.4 – P3.8
6-7	Chapter 4: Fundamental Data Types. The numeric data types, strings, characters, and booleans in Java. Variables, constants and literals of each of these primitive data types. The assignment statement, mathematical functions, type conversions, reading input, and comparison between primitive data types and objects. The String data type. Formatted I/O in Java. Suggested Exercises: R4.1 – R4.16, P4.1 – P4.3, P4.11
7-8	Chapter 5: Programming Graphics. How to write Java applications that display simple graphical shapes, and reading text input from this type of program. Suggested Exercises: R5.1- R5.12, P4.1, P4.2, P4.4, P4.6 – P4.9, P4.12, P4.15

⁵El tiempo separado para cada t3pico es algo aproximado, el mismo puede variar durante el transcurso del semestre dependiendo de las necesidades que surjan. Note que la secuencia de periodos solo incluye conferencias, no los ex3menes. Adem3s, el n3mero de periodos deja algunos periodos disponibles para ex3menes, repaso, discusi3n de ex3menes, proyectos, etc.

⁶ Los n3meros de los ejercicios corresponden a la 4ta edici3n de Java Concepts. Los n3meros correspondientes en la 5ta edici3n ser3n notificados pr3ximamente.

Day Number	Topic or Activity
9	Chapter 6: Decisions. We review the topic of decisions in programs, this time in the context of Java. We study selection constructs in Java, the <i>if-statement</i> and the <i>switch-statement</i>. We also study about Boolean expressions and grouping of statements in a program. Suggested Exercises: R6.1 – R6.19, P6.1, P6.5, P6.6, P6.8, P6.12, P6.18
10-11	Chapter 7: Iteration. We review the topic of iterations in programs in the context of Java. In particular, we study the <i>while</i> and <i>for loops</i> in Java. We study computer simulation as an example. Suggested Exercises: R7.1 – R7.13, P7.2, P7.6 - P7.10, P7.15, P7.16
EXAM	Exam 1
12-13	Chapter 8: Arrays and Array Lists. Introduction to arrays and array lists in Java. Wrapper classes, auto-boxing, the enhanced loop, and two-dimensional arrays. Different applications and typical operations with these structures are discussed. Suggested Exercises: R8.1 – R8.18, P8.1 – P8.6, P8.10, P8.16, P8.17
14-15	Chapter 9: Designing Classes. We discuss about the process of discovering classes during the analysis of program specifications, implementation by properly assigning attributes, pre-conditions and post-conditions, and definition of methods correctly. We will also study about static methods, variables, and how to use packages. Suggested Exercises: R9.1 – R9.32, P9.1 – P9.4, P9.10, P9.15, P9.16
16-17	Chapter 11: Interfaces and Polymorphisms. Interfaces in Java, their applications in defining reusable code, classes the implement interfaces. We also study the concept of supertype and subtype, and conversions between them, polymorphisms, the use and implementation of inner classes, and the implementation of listeners for timer events. Suggested Exercises: R11.1 – R11.17, P11.1 – P11.9, P11.13 – P11.15
18	Chapter 12: Event Handling. We will study one of the most fundamental concepts in the behavior of programs with graphical user interfaces, in which the user is to have control with the proper use of the mouse, clicking buttons, filing dialog boxes, specifying next actions to execute, hence controlling what the program does. We will study what events are and how they can be controlled through a Java Program. Suggested Exercises: R12.1 - R12.10, P12.1 – P12.5, P12.10, P12.12, P12.14, P12.15
19-20	Chapter 13: Inheritance. The concept of inheritance in object oriented programming and how it is achieved in Java. Suggested Exercises: R13.1 - R13.17, P13.1 – P13.6
EXAM	Exam 2
21	Chapter 14: Graphical User Interfaces. General issues concerning the use of a graphical user interface (GUI) for user to interact with a program in execution. We extend what we have learned so far by the use of graphics and events to be able to create more practical GUI's in which there are buttons, text components, combo boxes, and menus. This will be combined with the construction of programs that handle events from the user interface components. Suggested Exercises: R14.1 – R14.18, P14.1, P14.4 – P14.10
22	Chapter 15: Exceptions. We study the concept of program exceptions, how to create them in a Java program, and how, when and where to catch them in the program. Suggested Exercises: R15.1 – R15.14, P15.1 – P15.5

Day Number	Topic or Activity
23	Chapter 16: Streams. We will learn how to write Java programs that interact with disk files (I/O) and other sources of bytes and characters. We will also study data encryption as an application in which streams are useful (If time permits...) and serialization of objects. Suggested Exercises: R16.1 – R16.17, P16.1, P16.6 – P16.9
24	Chapter 17: Object-Oriented Design. Discussion of useful techniques and strategies to the design and implementation of software systems to handle non-trivial applications following the object-oriented approach. We discuss principles of UML, and relationship among classes. Suggested Exercises: R17.1 – R17.14, P17.8, P17.11
25-26	Chapter 18: Recursion. Deep study of the concept of recursion and the way it is implemented with the activation record and process stack. We will see applications in which recursion represents a natural solution, compare with iteration, and the use of helper or auxiliary recursive methods. Suggested Exercises: R18.1 – R18.10, P18.1, P18.2, P18.6, P18.7
EXAM	Exam 3
27-28	Chapter 19: Sorting and Searching. Sorting and searching are probably the most common tasks being performed in computer applications. Here we will study different algorithms to do both operations on arrays or lists. We will have a chance to study different algorithms to accomplish both tasks and compare them, hence having an introduction into the important field of “analysis of algorithms”. Suggested Exercises: R19.1 – R19.14, P19.1 – P19.4, P19.8, P19.12