

Universidad de Puerto Rico en Humacao  
Departamento de Matemáticas  
Programa de Bachillerato en Matemática Computacional

**A. Título del curso**

Introducción a la Programación y la Ciencia de Cómputos I

**B. Codificación del Curso**

COMP 3081

**C. Número de Horas/Crédito**

3 horas / 3 créditos

**D. Prerrequisitos, correquisitos y otros requerimientos**

A tomarse concurrentemente con el curso COMP3083 – Laboratorio de Introducción a la Programación y la Ciencia de Cómputos

**E. Descripción del Curso**

Fundamentos de la programación de dispositivos programables mediante un lenguaje de de alto nivel y sus implicaciones sociales y éticas. Variables de tipos primitivos, constantes, objetos, arreglos usados para representar, almacenar y encapsular información con los modos de asignación de valores a estos. Diseño e implantación de algoritmos simples mediante operaciones, estructuras de control de flujo, funciones, clases de biblioteca, enunciados compuestos. Se programa interfaces textuales e interfaces gráficos sencillos. El estudiante hará durante la clase ejercicios cortos y frecuentes relacionados a los temas discutidos en un dispositivo programable propio o de la institución.

**F. Objetivos del Curso**

Objetivos Generales:

Al finalizar el curso el estudiante:

1. Conocerá el origen, la historia, aspectos sociales y los fines de la programación.
2. Desarrollará algoritmos sencillos para la solución de problemas.
3. Escribirá programas utilizando formas de expresión y estilo convencionales entre profesionales de la informática.
4. Programará una computadora u otro dispositivo programable para uso general en un lenguaje de alto nivel utilizando conceptos fundamentales de la programación.
5. Resolverá problemas que requieren la escritura de código incorporen instrucciones de asignación, estructuras de control de ejecución, uso de objetos de biblioteca, etc.

Objetivos específicos:

Al finalizar el curso el o la estudiante estará capacitado para<sup>1</sup>:

---

1 Se clarifica entre paréntesis cuadrados [] el nivel de habilidad cognitiva a alcanzar en el objetivo. Estas son, en orden ascendente de nivel, *conocimiento, comprensión, aplicación, análisis, síntesis y evaluación*. La nomenclatura sigue las

### I. Contexto Histórico y Social

1. Nombrar al menos cinco hitos significativos y las personas asociadas a ellos en la historia de la ciencia de cómputos. [Conocimiento]
2. Discutir cómo el contexto histórico influye sobre el desarrollo de la arquitectura de computadoras y los lenguajes de programación para al menos dos épocas. [Comprensión]
3. Reconocer al menos un cambio significativo que ha provocado el desarrollo de las computadoras y el INTERNET en la cotidianidad de la gente. [Conocimiento]
4. Identificar al menos dos instancias de discriminación por razones ajenas al mérito técnico y/o académico en el acceso a las tecnologías de computación y su desarrollo. [Comprensión]

### II. Introducción a la programación en un lenguaje de alto nivel

1. Utilizar todos los tipos de datos primitivos y compuestos provistos por el lenguaje de alto nivel usado para impartir el curso. [Aplicación]
2. Explicar cómo los tipos estáticos ayudan en la detección de errores antes de que el programa sea ejecutado. [Comprensión]
3. Analizar el comportamiento y modificar programas que utilicen constantes y variables de tipos primitivos, y objetos de clases de biblioteca. [Análisis]
4. Construir al menos dos programas que declaren constantes y variables de tipos primitivos, objetos de clases de biblioteca y las usen en expresiones y (re)asignaciones de valor. [Síntesis]

### III. Control de ejecución

1. Analizar el comportamiento y modificar al menos dos programas que utilicen las estructuras de control de ejecución condicionales, ciclos, llamadas a funciones y métodos de clases de biblioteca. [Comprensión]
2. Trazar la ejecución de al menos seis segmentos de código que utilicen las estructuras de control de ejecución condicionales, ciclos, llamadas a funciones y métodos de clases de biblioteca individualmente o en combinación en función de determinar su corrección. [Evaluación]
3. Discriminar entre todas las estructuras de control de ejecución condicionales o iterativas provistas para el lenguaje apropiadas para un objetivo particular. [Análisis]
4. Usar arreglos para guardar, ordenar y manipular al menos dos tipos de conjuntos de datos. [Aplicación]
5. Pasar información a funciones y métodos y recibir sus resultados mediante argumentos, parámetros y valores de retorno. [Aplicación]
6. Representar y manipular información valiéndose de los conceptos de tipos de datos, constantes, variables, objetos y las operaciones y métodos definidos para ello. [Aplicación]

### IV. Diseño de algoritmos

1. Discutir el papel del algoritmo en el software y enunciar al menos dos ejemplos de cómo una misma tarea puede ser hecha con muchos algoritmos diferentes. [Comprensión]

2. Esbozar algoritmos en un lenguaje natural (ejemplo Español, Inglés o pseudo-código). [Aplicación]
  3. Re-escribir un algoritmo escrito en lenguaje natural usando un lenguaje de programación de alto nivel. [Aplicación]
  4. Identificar los objetos, métodos y algoritmos que están implícitamente mencionados en una especificación de una tarea. [Comprensión]
  5. Diseñar, implantar, probar y depurar un programa que utilicen estructuras fundamentales de programación combinando código propio, clases y funciones de biblioteca, y/o código de dominio público modificado. [Análisis]
  6. Describir el concepto de recursión, su estructura básica de casos base y general, y aplicarlo en la solución problemas cuando sea apropiado. [Aplicación]
- V. Encapsulación, protección de información y programación orientada a objetos.
1. Escribir clases que encapsulen datos y los métodos que los manipulan. [Síntesis]
  2. Diseñar e implantar clases con al menos dos campos de información. [Síntesis]
  3. Ilustrar los mecanismos provistos por el lenguaje para la determinación dinámica de la ejecución de mensajes enviados a objetos.[Comprensión]
  4. Definir los conceptos de protección de información, acoplamiento, cohesión, y encapsulación. [Conocimiento]
  5. Explicar cómo las referencias a objetos permiten su acceso por diferentes vías. [Comprensión]
- VI. Desarrollo de programas
1. Escribir todos los programas en forma modular y encapsulando datos y los métodos que los manipulan mediante clases cuando sea apropiado. [Síntesis]
  2. Describir uno o mas patrones de diseño que sean aplicables a un sistema de software simple: estructurado/funcional vs. Orientado a objetos. [Comprensión]
  3. Explicar la importancia del desarrollo centrado en el humano del software. [Comprensión]
  4. Ilustrar con al menos cuatro ejemplos cómo errores comunes en la codificación que llevan a programas inseguros y cómo aplicar técnicas para evitarlos. [Comprensión]
  5. Escribir todos los programas utilizando formas de expresión y estilo convencionales entre profesionales de la informática. [Aplicación]

Algunos de estos objetivos específicos han sido adaptados de las recomendaciones contenidas en el *Computer Science Curricula 2013* (ver bibliografía).

## **G. Bosquejo del Contenido y Distribución de Tiempo**

- I. Contexto Histórico y Social (4 horas)
  1. Artefactos programables previo al 1946
  2. Invención y desarrollo de los artefactos programables de uso general (computadoras), su arquitectura, el software y las redes
  3. Pioneros y pioneras de la computación

4. Diferencias de acceso a la tecnología digital por razón de género, etnia, clase social, entre otros
5. Historia del Internet
- II. Introducción a la programación en un lenguaje de alto nivel (12 horas)
  1. Sintaxis básica
  2. Literales, variables y tipos primitivos
  3. Expresiones y asignaciones de valor
  4. Compilado. Errores de sintaxis, ejecución y lógicos
  5. Cadenas de caracteres y arreglos
  6. Uso de funciones y clases predefinidas en bibliotecas
  7. Tipos de interacción con el(la) usuario: textual, gráfica, etc)
  8. Asociación entre tipos y variables, argumentos, resultados y campos
  9. Interfaces gráficos de usuario: componentes básicos, metáforas gráficas, y nociones básicas de color, percepción y ergonomía
  10. Fuentes de información sobre diseño de interacción
- III. Control de ejecución (8 horas)
  1. El papel del algoritmo en el proceso de solución de problemas
  2. Estructuras de control de flujo: toma de decisiones y repetición
  3. Procesamiento de cadenas de caracteres y arreglos
  4. Funciones, métodos, paso de parámetros, variables locales vs. Variables globales
- IV. Diseño de algoritmos (5 horas)
  1. Refinado de algoritmos
  2. Refactorización simple
  3. Comprensión de programas
  4. Algoritmos fundamentales de ordenamiento y búsqueda. Recursión
  5. Medición informal de eficiencia de algoritmos (conteo de operaciones)
- V. Encapsulación, protección de información y programación orientada a objetos (6 horas)
  1. Manera básica de definir clases. Relación entre objeto y clase
  2. Campos para almacenar información en una clase
  3. Funciones como métodos, tipos de métodos (constructores, destructores, modificadores del estado del objeto, extractores de información del objeto, Iteradores)
  4. Subclases, herencia, anulación de método (overriding) en clases de biblioteca
- VI. Desarrollo de programas (5 horas)
  1. El proceso de escribir un programa:
    - (a) Análisis y definición

- (b) Diseño
  - (c) Implantación
  - (d) Depuración
  - (e) Documentación y mantenimiento
2. Principios de diseño de software: modelos estructurales y de comportamiento, patrones de diseño
  3. Ambientes de programación avanzados
  4. Bibliotecas y sus Interfaces de Programación de Aplicaciones (APIs)
  5. Documentación y estilos de programación

La suma de las horas sugeridas es de 40. Las cinco horas restantes serán utilizadas para la ejecución de las estrategias de instruccionales afines con el curso (ver sección H). Los tópicos en este bosquejo de contenido son aptos para ser ordenados de otras maneras y actualizados a juicio del docente que imparta el curso en el ejercicio de su libertad de cátedra.

#### **H. Técnica Instruccionales**

Con miras a lograr los objetivos del curso el o la profesora seleccionará entre las siguientes técnicas instruccionales: conferencias, demostraciones, laboratorios, trabajos en grupo, estudios independientes, discusión de asignaciones. El trabajo de laboratorio es parte integral del curso.

#### **I. Recursos de aprendizaje e instalaciones mínimas disponibles o requeridos**

La Universidad debe proveer un laboratorio para trabajo independiente y actividades en clase de los estudiantes, el equipo electrónico que necesita el profesor para impartir la clase, el programado apropiado para el curso (Intérpretes y Compiladores de todos los lenguajes que se utilizarán, herramientas para el desarrollo de aplicaciones) y acceso a INTERNET. Se deberá proveer acceso a red y asistencia en la configuración de los compiladores, intérpretes u otros programados utilizados en este curso en la computadora portátil que debe poseer cada estudiante.

#### **J. Estrategias de Evaluación**

Dos o más exámenes parciales y un examen final, para un 60% de la nota; otros trabajos como asignaciones de programación, monografías y pruebas compondrán 10%. La calificación obtenida en el curso de laboratorio COMP 3083 que el(la) estudiante tomará concurrentemente representa el 30% de la nota.

#### **K. Acomodo razonable**

La UPR-Humacao cumple con las ley ADA (“*American with Disabilities ACT*”) y la ley 51 (Servicios Educativos Integrales para Personas con Impedimento) para garantizar igualdad en el acceso a la educación y servicios. Estudiantes con impedimentos: informe al (la) profesor(a) de cada curso sobre sus necesidades especiales y/o de acomodo razonable para el curso, en la tarjeta de información que llena en la primera semana de clases y visite la Oficina de Servicios para la Población con Impedimentos (SERPI) a la brevedad posible. Se garantiza confidencialidad.

#### **L. Integridad académica**

La Universidad de Puerto Rico promueve los más altos estándares de integridad académica y

científica. El Artículo 6.2 del Reglamento General de Estudiantes de la UPR (Certificación Núm. 13, 2009-2010, de la Junta de Síndicos) establece que “*la deshonestidad académica incluye, pero no se limita a: acciones fraudulentas, la obtención de notas o grados académicos valiéndose de falsas o fraudulentas simulaciones, copiar total o parcialmente la labor académica de otra persona, plagiar total o parcialmente el trabajo de otra persona, copiar total o parcialmente las respuestas de otra persona a las preguntas de un examen, haciendo o consiguiendo que otro tome en su nombre cualquier prueba o examen oral o escrito, así como la ayuda o facilitación para que otra persona incurra en la referida conducta*”. Cualquiera de estas acciones estará sujeta a sanciones disciplinarias en conformidad con el procedimiento disciplinario establecido en el Reglamento General de Estudiantes de la UPR vigente.

#### **M. Sistema de Calificación**

Se adjudicará la nota A, B, C, D, ó F según el nivel de competencia demostrado en las evaluaciones.

El profesor o profesora informará, las primeras semanas de clases (y en la guía de estudiante), la curva a usarse para asignar las calificaciones.

#### **N. Política sobre discriminación.**

La Universidad de Puerto Rico no discrimina por razones de edad, sexo, raza, color, nacionalidad, origen o condición social, ni por ideas políticas, religiosas e impedimentos físicos o mentales o por condición de veteranos y está comprometida con el acomodo razonable de estudiantes y empleados. Patrono con igualdad de oportunidades en el empleo.

#### **O. Bibliografía**

1. ACM/IEEE-CS Joint Task Force on Computing Curricula (2013). *Computer Science Curricula 2013*. Technical report, ACM Press and IEEE Computer Society Press. ISBN 978-1-4503-2309-3
2. Schildt, H (2006). *Java: The Complete Reference* (9th ed.). New Jersey: McGraw-Hill Osborne Media. ISBN-13: 978-0071808552
3. Deitel, P. J., & Deitel, H. M. (2015). *Java: How to program: Early objects version* (10th ed.). Toronto: Pearson Prentice Hall. ISBN: 0133807800
4. Deitel, P. J., & Deitel, H. M. (2015). *Java: How to program: Late objects version* (10th ed.). Toronto: Pearson Prentice Hall. ISBN: 0132575655
5. Deitel, P. J., & Deitel, H. M. (2011). *C++ how to program* (8th ed.). upper saddle river: Pearson Education. ISBN
6. Misa, T. J. (2010). *Gender codes why women are leaving computing*. John Wiley and Sons. ISBN 978-0470-59719-4
7. Gaddis, T. (2010). *Starting out with Java: from control structures through objects* (5th ed.). Pearson Education. ISBN 978-0132855839 978-0133769395
8. Kernigan, B. & Ritchie, D. (1978). *The C programming language*. Prentice-Hall. ISBN 0-13-110163-3 (libro clásico)
9. Levy, S. (2010). *Hackers: Heroes of the Computer Revolution*. (25th ed.). New York: O'Reilly Media, Incorporated. ISBN 978-1449388393

10. Olson, P. (2012). *We are Anonymous: inside the hacker world of Lulzsec, Anonymous, and the global cyber insurgency*. Little, Brown and Company. ISBN 978-0316213523
11. Sotero Esteva, J. O. (2012). *Introducción al Desarrollo de Aplicaciones Usando Java*. Edición beta 0.04. Manuscrito sin publicar. Universidad de Puerto Rico en Humacao, Humacao, Puerto Rico. (disponible en el Departamento de Matemáticas, UPRH)
12. Gaddis, T. (2014). *Starting Out with C++: From Control Structures through Objects*. (8<sup>th</sup> Ed.) Addison-Wesley Publishing Company. ISBN 978-0133769395

Rev/MLV/jsm/PD/ProntuarioCOMP3081/25-jun.-07

Rev/Bibliográfica/Profa. B. Santiago-Figueroa, Oct. 05

Rev/Comité de Currículo, Oct. 05

Aprob/DM/15-nov.-05

Revisión por José O. Sotero Esteva, 11 de octubre de 2011.

Revisión por José O. Sotero Esteva con el insumo de Idalyn Ríos, Jorge Montoya y Elio Ramos Colón, octubre de 2014.

Esta obra está licenciada bajo la *Licencia Creative Commons Atribución 4.0 Internacional*. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>. Se provee la cita:

Sotero Esteva, J.O., Ríos, I., Montoya, J., Ramos Colón, E. (2014). *Prontuario Introducción a la Programación y la Ciencia de Cómputos I*. Departamento de Matemáticas, Universidad de Puerto Rico en Humacao.