

Programación matemática 1

1. Descripción del Curso

Nombre: Programación Matemática 1	Código: M404
Prerrequisitos: M303	Créditos: 5
Profesor: Hugo García	Semestre: Segundo, 2018

Curso enfocado al aprendizaje de programación de computadoras usando como lenguaje python. Se desarrolla programas informáticos empleando diversos paradigmas de programación.

2. Competencias

2.1. Competencias generales

- 2.1.1 Capacidad para formular problemas, tomar decisiones e interpretar las soluciones en los contextos originales de los problemas.
- 2.1.2 Capacidad para utilizar las herramientas computacionales para plantear y resolver problemas.
- 2.1.3 Capacidad para comprender problemas, abstraer lo esencial de ellos y resolverlos.
- 2.1.4 Capacidad para extraer información cualitativa de datos cuantitativos.
- 2.1.5 Capacidad para detectar inconsistencias.

2.2. Competencias específicas

- a. El estudiante estudia problemas de la vida real, abstrae y diseña un algoritmo para resolverlo.
- b. El estudiante implementa algoritmos en código python para la resolución de problemas.
- c. El estudiante usa metodología UML para implementar sus soluciones.
- d. El estudiante depura su programa para verificar la fiabilidad del software.
- e. El estudiante utiliza interfaces gráficas de usuarios para presentar sus programas informáticos.

3. Unidades

3.1. Bases e historia de la computación

Descripción: Panorama de ciencia de la computación. La computación desde una perspectiva histórica. Lenguajes de programación y su historia. Lenguajes compilados vs intérpretes. Sistemas operativos. La arquitectura de Von Neumann. Funciones de conmutación, diseño lógico. Redes de computadoras.

Duración: 20 períodos de 50 minutos

Metodología: Los períodos de clase son magistrales.

Evaluación: Se evaluará a través de tareas semanales y preguntas directas en exámenes.

3.2. Algoritmos y programación en python

Descripción: Algoritmos y su complejidad. Ciclos en python. Decisiones en python. Tipos de datos elementales en python. Paradigma de orientación a objetos. Programación con clases. Entrada salida de archivos. Principios de ingeniería de software. El software como obra de arte vs producto de ingeniería. Herencia. Polimorfismo. Clases abstractas. Excepciones. Depuración. Ciclo de vida del software. Metodología de diseño UML. Elementos del lenguaje. Programación en html, css, javascript y latex.

Duración: 56 períodos de 50 minutos

Metodología: Los períodos de clase son magistrales dando ejemplos de implementación de algoritmos en lenguaje de programación python.

Evaluación: Se evaluará a través de tareas semanales, preguntas directas, ejercicios semanales de resolución de problemas de el sitio web <http://www.projecteuler.net> y tres problemas en el primer examen parcial.

4. Evaluación del curso

Los porcentajes asignados a cada uno de los elementos de la evaluación están de acuerdo con el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala

Proyecto	30 puntos
Tareas semanales	5 puntos
Práctica 1	7 puntos
Práctica 2	8 puntos
Exámenes parciales	25 puntos
Examen final	25 puntos
Total	100 puntos

5. Bibliografía

1. Cuevas, Alberto. "Python 3 Curso Práctico". Bogotá: Ediciones de la U. 2017
2. Downey, Allen. "How to think like a computer scientist". EEUU. 2002
3. Joyanes Luis. "Fundamentos de programación". Cuarta ed. 2008.
4. Zelle, Jhon. "Python Programming: An Introduction to computer science". EEUU. 2002
5. <http://learnpythonthehardway.org/book/>
6. Gauchat, Juan. "El gran libro de HTML5, CSS3 y Javascript". España. 2012.

<http://ecfm.usac.edu.gt/programas>