



EE463 - Programación II

Nombre del programa académico	Ingeniería electrónica
Nombre completo de la asignatura	Programación II
Área académica o categoría	Básicas de ingeniería electrónica
Semestre y año de actualización	2023-2
Semestre y año en que se imparte	2023-2
Tipo de asignatura	☒ Obligatoria ☐ Electiva
Número de créditos	3
Director o contacto del programa	Arley Bejarano Martínez
Coordinador o contacto de la asignatura	Arley Bejarano Martínez

Descripción y contenidos

1. Breve descripción Es una asignatura teórico-práctica, la cual tiene como finalidad la enseñanza de metodologías de programación orientada a objetos y a eventos, esto para la implementación de algoritmos estandarizados incluyendo el desarrollo de la habilidad del diseño e implementación de interfaces gráficas.
2. Objetivos • Analizar, estructurar, diseñar e implementar soluciones de cómputo utilizando paradigmas de programación como el orientado a objetos y a eventos. • Utilizar lenguajes de programación estructurados como C, C++ y Python. • Diseñar e implementar interfaces gráficas utilizando paradigmas de programación como el orientado a objetos y a eventos.
3. Resultados de aprendizaje • RAA1: Estructura, diseña e implementa soluciones a problemas de cómputo utilizando programación orientada a objetos. Se corresponde con los RAP 1 y 2 • RAA2: Desarrolla código fuente utilizando lenguajes de programación como C, C++ y Python. Se corresponde con los RAP 1 y 2 • RAA3: Estructura, diseña e implementa soluciones a través de interfaces gráficas, esto utilizando paradigmas de programación como el orientado a objetos y a eventos. Se corresponde con los RAP 1 y 2
4. Contenido • Programación modular en C y C++ (Paso por valor, paso por puntero y referencia). (8 H) • Manejo de memoria y tópicos especiales. (4 H) • Introducción a la programación orientada a objetos en C y C++. (8 H) • Herencia y Polimorfismo en C y C++. (4 H) • Programación estructurada en Python (If- Else, While, For, Switch-Case y Arreglos) (4 H) • Funciones y estructuras en Python (8H) • Programación orientada a objetos en Python y archivos tipo Header's. (8 H) • Gestión de proyectos y control de versiones con GitHub (4H) • Diseño e implementación de interfaces gráficas (16 H)
5. Requisitos:



Programación I (EE373)
6. Recursos - Fatos Xhafa, Pere Pau Vasquéz, Jordi Marco, Xavier Molinero y Ángela Martín. Programación en C++ para Ingenieros. International Thomson Editores Spain Paraninfo S.A. 1ra Edición. - Cplusplus. C++ Language. Disponible en: https://cplusplus.com/doc/tutorial/. - Geeksforgeeks. C++ Programming Language. Disponible en: https://www.geeksforgeeks.org/c-plus-plus/?ref=shm_outind - W3schools. Python Tutorial. Disponible en: https://www.w3schools.com/python/default.asp - Mathworks. Object-Oriented Programming in MATLAB. Disponible en: http://www.mathworks.com/discovery/object-oriented-programming.html. - Geeksforgeeks. Python Tutorial. Disponible en: https://www.geeksforgeeks.org/python-programming-language/ - Geeksforgeeks. Python GUI – tkinter. Disponible en: https://www.geeksforgeeks.org/python-gui-tkinter/ - Software DEV++ - Software Pycharm y Anaconda. - Software Matlab.
7. Herramientas técnicas de soporte para la enseñanza - Clase magistral. - Estudio y análisis de casos prácticos. - Realización de trabajos individuales. - Talleres para solución en grupo sobre cada contenido de la materia. - Consultas utilizando la bibliografía del curso. - Proyecto integrador de asignaturas desarrollando software para solucionar problemas de ingeniería. - Uso de videotutoriales para apoyar los conceptos vistos en clase. - Creación de sitio web o aula virtual para la comunicación con estudiantes y almacenamiento de material de clase.
8. Trabajos en laboratorio y proyectos - Esta materia contempla 2 horas de práctica en la semana en la que se ejercicios para el desarrollo de habilidades según el contenido abordado durante las horas teóricas de la asignatura. - Esta materia contempla un proyecto final de curso en el que el estudiante debe utilizar los conceptos aprendidos en clase para resolver un problema real.
9. Métodos de aprendizaje - Este curso se desarrollará con la interacción directa estudiante-profesor en la explicación de los conceptos de la teoría para el desarrollo de programas de computador apoyándose con asignación de trabajo extra-clase, talleres de clase. - Como una herramienta de refuerzo de contenidos, se realizará la asignación de material complementario, para lograr el entendimiento del contenido, así como el refuerzo de los resultados de aprendizaje. - Se realizará el diseño e implementación de software que resuelva un problema real elegido por los estudiantes estimulando de manera activa la participación del estudiante a partir de interrogantes que generan tanto la creatividad, como el pensamiento



10. Métodos de evaluación

Teniendo en cuenta el Acuerdo 29 de 2006 del Consejo Académico las evaluaciones se llevarán a cabo en los siguientes momentos:

- Evaluación Parcial 1 (Porcentaje de la materia: 30%): Se plantea un problema en el cual los estudiantes deben aplicar los conceptos de POO, esto a partir del lenguaje de programación C++. Su objetivo se basa en afianzar el conocimiento de los alumnos con respecto a los temas orientados en clase. Esta evaluación se realiza antes de la semana 8 del periodo académico. Se evalúa el resultado de aprendizaje: RAA-1 y RAA-2.
- Evaluación Parcial 2 (Porcentaje de la materia: 30%): Se propone un problema en el cual los estudiantes deben aplicar los conceptos de POO utilizando el lenguaje de programación Python. Su objetivo se basa en afianzar el conocimiento de los alumnos con respecto a los temas orientados en clase. Esta evaluación se realiza en el transcurso de la semana 12. Se evalúa el resultado de aprendizaje: RAA-1 y RAA-2.
- Evaluación Parcial 3 (Porcentaje de la materia: 20%): Se plantea un examen en el cual se propone desarrollar una interfaz gráfica en Tkinter utilizando los paradigmas de programación orientada a objetos y a eventos, esto con la finalidad de dar solución a un problema específico. Esta evaluación se realiza en el transcurso de la semana 15 o 16. Se evalúa el resultado de aprendizaje: RAA-3.
- Evaluación proyecto Final (Porcentaje de la materia 20%): Proyecto de desarrollo de software en donde el estudiante debe resolver un problema de la sociedad o la industria utilizando interfaces gráficas. Esta evaluación se realiza en la semana de finales. Se evalúa el resultado de aprendizaje RAA-3.

11. Estrategias para la enseñanza y el aprendizaje

- Utilizar ejemplos concretos tanto de manera teórica como sintáctica, en donde se presentan con claridad los conceptos orientados durante las clases.
- Fomentar la participación de los estudiantes durante las clases a través de preguntas y ejercicios interactivos.
- Proporcionar herramientas extra-clases para el estudio de los conceptos orientados en clase.
- Proponer actividades en grupo para fomentar el trabajo en equipo colaborativo.