

TALLER DE PROGRAMACIÓN DE COMPUTADORAS

- Propósito** Promover la indagación científica y el diseño en ingeniería aprendiendo a programar computadoras con varios paradigmas y lenguajes de programación.
- Metodología** Aprendizaje basado en proyectos con instrucción 5E y evaluación Bloom. Actividades reflexivas y empleo de mapas conceptuales, rúbricas y portafolios. Clases presenciales o por Internet. Dos horas adicionales de estudio personal por hora de clase.
- Horario** Acordado con los participantes. Cuatro fases de cinco sesiones. El temario puede ser ajustado para un taller de 20 o de 40 horas de clases.
- Soporte tecnológico** Entornos de programación para C++, Python y Octave de fuentes gratuitas. Bibliografía, lecturas y referencias de acceso público.

PROGRAMA ACADÉMICO

I. Fundamentos de programación

Estado del arte y tendencias de la programación de computadoras y sus aplicaciones.

1. Programas, algoritmos, software y códigos. Ingeniería de software. Procesamiento de datos. *Introducción a Python.*
2. Algoritmos, estructuras de datos y operadores. Selección y repetición de rutinas. Revisión de álgebra lineal. *Introducción a Octave.*
3. Programación orientada a objetos. Clases, objetos, métodos, herencia y polimorfismo. *Introducción a C++.*
4. Programación de microcontroladores. Sensores y actuadores. *Experimento de cinemática usando el lenguaje C y la tarjeta Arduino Uno.*
5. Verificación, depuración y documentación de programas. Oportunidades en computación y programación.

II. Ejemplos de Programación

Método de instrucción 5E para analizar algoritmos, estructuras de datos y códigos de programas.

6. Ordenamiento y búsqueda de datos. *Edición de archivos en Python.*
7. Análisis de señales digitales. *Transformadas y filtros en Octave.*
8. Enseñanza mediante computador. *Simulador de circuitos lógicos en C++.*
9. Caso de estudio: *juego de cuatro dígitos.*
10. Caso de estudio: *navegador web.*

III. Diseño de Programas

Proceso de diseño en ingeniería para crear programas que cumplan especificaciones técnicas.

11. Experimento virtual de cinemática.
12. Editor de circuitos electrónicos.
13. Respuesta transitoria de sistemas lineales.
14. Supervisor remoto de señales climatológicas.
15. Reconocimiento de objetos.

IV. Proyecto de Ingeniería

Aplicación de los entendimientos y habilidades de análisis y diseño de programas para resolver problemas de computación del mundo real.

16. El proceso de diseño en ingeniería. Entrevistas con personas. Evaluaciones de oportunidades de programación. Presupuesto y cronograma del proyecto.
17. Selección del lenguaje y bibliotecas de soporte. Diseño de la arquitectura y de la verificación del software. Prueba y depuración de los programas.
18. Integración de los programas y evaluación del software. Entrevistas con los usuarios para evaluar las funciones solicitadas.
19. Reflexión, planificación de las revisiones y mejoras del software. Redacción de los documentos de soporte.
20. Exposición pública para demostrar tanto los entendimientos y habilidades de programación como el software desarrollado.

Más información: Ing. Arturo J. Miguel de Priego Paz Soldán

www.tourdigital.net

Correo personal: amiguel@puce.edu.pe