

Misión del Centro Universitario

Somos un centro que forma parte de la Red Universitaria de la Universidad de Guadalajara. Como institución de educación superior pública asumimos el compromiso social de satisfacer necesidades de formación y generación de conocimiento en el campo de las ciencias exactas y las ingenierías. La investigación científica y tecnológica, así como la vinculación y extensión, son parte fundamental de nuestras actividades para incidir en el desarrollo de la sociedad; por lo que se realizan con vocación internacional, humanismo, calidad y pertinencia.

1.- Identificación de la Unidad de Aprendizaje

Nombre de la Unidad de Aprendizaje

Programación Aplicada

Clave de la UA	Modalidad de la UA	Tipo de UA		Valor de créditos	Área de formación
17404	Presencial	Curso, Taller		7	Básica común (INME) Optativa abierta (INDU)
Hora semana		Horas teoría/semestre	Horas práctica/ semestre	Total de horas:	Seriación
4		34	34	68	Ninguno
Departamento			Academia		
Departamento de Ciencias Computacionales			Programación		

Presentación

En este curso se estudiarán los conceptos básicos de programación, haciendo especial énfasis en comprender el proceso de traducción del planteamiento de un problema a su formulación computacional para resolverlo. El estudiante utilizará el lenguaje de programación Python para resolver problemas computables.

Competencia de la Unidad de Aprendizaje (UA)

Elabora soluciones a problemas sencillos mediante un lenguaje de programación que resuelven el problema correctamente.

<p>Tipos de saberes</p> <p>Se refiere al desglose de aquellos conocimientos, habilidades, actitudes y valores que se encuentran ligados a la descripción de la competencia, y al desarrollarlos deben observar la parte de los nuevos aprendizajes y capacidades que logrará el estudiante</p>		
Saber (conocimientos)	Saber hacer (habilidades)	Saber ser (actitudes y valores)
<p>Esencia operativa de las computadoras.</p> <p>Nociones básicas del acto de programar computadoras.</p> <p>Elementos básicos de un lenguaje de programación. Construcción de algoritmos para solucionar problemas sencillos.</p> <p>Funcionamiento de un programa a partir de su código fuente.</p>	<p>Abstracción.</p> <p>Manejo de Razonamiento lógico.</p> <p>Razonamiento matemático.</p> <p>Análisis y síntesis de información.</p> <p>Destreza en el uso de la computadora.</p> <p>Lectura y comprensión de idioma inglés.</p>	<p>Es responsable de en las fechas de entrega.</p> <p>Colabora con sus compañeros para mejorar el trabajo en equipo.</p> <p>Autogestión.</p> <p>Honestidad en el desarrollo de las actividades académicas.</p> <p>Responsabilidad y compromiso en las actividades individuales y por equipo.</p> <p>Toma decisiones y se proyecta como líder.</p>
Competencia genérica		Competencia profesional
<p>Pensamiento matemático</p> <p>Pensamiento crítico y reflexivo</p> <p>Aprendizaje autónomo</p>		Diseño y desarrollo de software
Competencias previas del alumno		
<ul style="list-style-type: none"> ● Escucha, interpreta y emite mensajes pertinentes en distintos contextos mediante la utilización de medios, códigos y herramientas apropiados ● Desarrolla innovaciones y propone soluciones a problemas a partir de métodos establecidos. ● Sustenta una postura personal sobre temas de interés y relevancia general, considerando otros puntos de vista de manera crítica y reflexiva. ● Aprende por iniciativa e interés propio a lo largo de la vida. ● Participa y colabora de manera efectiva en equipos diversos. 		
Competencia del perfil de egreso		
Diseña y desarrolla software		
Perfil deseable del docente		
Docente con licenciatura en ingeniería en computación o carrera afín, con conocimientos en programación, y conocimientos del lenguaje de programación Python.		

2.- Contenidos temáticos

Contenido

1.1 Conceptos básicos de programación estructurada	29 HORAS
1.1.1 Introducción a la computación	
1.1.2 Definición y características de programación estructurada	
1.1.3 Elementos básicos de un programa estructurado	
1.2 Algoritmo	
1.2.1 Definición de algoritmo	
1.2.2 Técnicas para la formulación de algoritmos	
1.2.2.1 Diagramas de flujo	
1.2.2.2 Pseudocódigo	
1.3 Estructuras de control	
1.3.1 Definición	
1.3.2 Secuencial	
1.3.3. Selectiva	
1.3.3.1 Si y si-sino	
1.3.3.2 Según sea	
1.3.4 Estructuras de control repetitivas	
1.3.4.1 Contadores, acumuladores y banderas	
1.3.4.2 Mientras	
1.3.4.3 Hacer - mientras	
1.3.4.4 Desde	
1.3.5 Estructuras anidadas	
1.3.6 Definición	
1.3.7 Implementación.	

2. Arreglos

12 HORAS

2.1 Definición

2.2 Tipos de arreglos

2.2.1 Arreglos Unidimensionales (1 dimensión)
vectores

2.2.2 Arreglos Bidimensionales (2 dimensiones)
matrices

3. Manejo de funciones

11 HORAS

3.1 Definición

3.2 Funciones sin paso de parámetros

3.3 Funciones con parámetros por valor

3.3.1 Funciones con parámetros usando datos de
tipo primitivo

3.3.2 Funciones con parámetros usando datos de
tipo arreglo

4. Colecciones pre fabricadas en Python

11 HORAS

4.1 Definición

4.2 Operaciones con una lista, tupla, conjunto,
diccionario.

4.2.1 Asignación a los datos de una lista, tupla,
conjunto, diccionario

4.2.2 Salida de datos de una lista, tupla, conjunto,
diccionario

4.3 Arreglos con elementos de tipo lista.

4.3.1 Definición

4.3.2 Operaciones con arreglos de tipo lista y
funciones.

4.3.2.1 Entrada de datos de los elementos de un
arreglo de lista con funciones desarrolladas
por el usuario.

4.3.2.2 Salida de datos de un arreglo de listas y
funciones.

Estrategias docentes para impartir la unidad de aprendizaje
<ol style="list-style-type: none"> 1. Método expositivo 2. Resolución de ejercicios y problemas 3. Aprendizaje orientado a proyectos 4. Realizar prácticas 5. Trabajo en equipo.
Bibliografía básica
<p>Guttag, J.V., Introduction to Computation and Programming Using Python, 2013, MIT Press.</p> <p>Conery, J., Explorations in Computing: An Introduction to Computer Science and Python Programming, 2014, Chapman and Hall/CRC.</p> <p>Johansen, A., Python: The Ultimate Beginner's Guide!, 2016, CreateSpace Independent Publishing Platform.</p> <p>Matthes, E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming, 2015, No Starch Press.</p>
Bibliografía complementaria
Sedgewick, R., and Wayne, K., and Dondero, R. Introduction to Programming in Python: An Interdisciplinary Approach, 2016, Addison-Wesley Professional.
3.-Evaluación
Evidencias
<p>Reporte escrito que contenga las definiciones de los conceptos básicos indicando la fuente de información utilizada.</p> <p>Reportes que contengan la solución de los problemas que les fueron planteados y éstos deberán incluir:</p> <p>Descripción del problema.</p> <p>Diagrama de flujo.</p> <p>Pseudocódigo.</p> <p>Corrida de escritorio de la solución propuesta.</p> <p>Código fuente</p>

Reporte escrito que contenga las definiciones de las estructuras selectivas. indicando la fuente de información utilizada.

Reportes que contengan la solución de los problemas que les fueron planteados y éstos deberán incluir:

- Descripción del problema.
- Diagrama de flujo.
- Pseudocódigo.
- Corrida de escritorio de la solución propuesta.
- Código fuente

Reporte escrito que contenga las definiciones de las estructuras repetitivas. indicando la fuente de información utilizada.

Reportes que contengan la solución de los problemas que les fueron planteados y éstos deberán incluir:

- Descripción del problema.
- Diagrama de flujo.
- Pseudocódigo.
- Corrida de escritorio de la solución propuesta.
- Código fuente

Reporte escrito que contenga las definiciones de los diferentes tipos de arreglos, indicando la fuente de información utilizada.

Reportes que contengan la solución de los problemas que les fueron planteados y éstos deberán incluir:

- Descripción del problema.
- Corrida de escritorio de la solución propuesta.
- Código fuente en python.

Examen parcial

Solicita a los estudiantes lecturas previas acerca de Programación Modular (funciones)

Plantea una serie de ejercicios donde se apliquen funciones

Reporte escrito que contenga las características de los registros, indicando la fuente de información utilizada.

Reportes que contengan la solución de los problemas que les fueron planteados y éstos deberán incluir:

- Descripción del problema.
- Corrida de escritorio de la solución propuesta.

Código fuente

Examen Parcial

Tipo de evaluación

Heteroevaluación procedimental
Criterios de Evaluación (% por criterio)
Examen global final 20 % Actividades realizadas 90 %
4.-Acreditación
Tener por lo menos el 80% de asistencia a clases Obtener calificación aprobatoria en la unidad de aprendizaje
Tener por lo menos 65% de asistencia a clases Obtener calificación aprobatoria en el examen extraordinario
5.- Participantes en la elaboración
Fecha de revisión: