



Nombre de la asignatura									Programación I	Clave de la asignatura C0108006
Área de formación	Docencia frente a grupo según SATCA				Trabajo de Campo Supervisado según SATCA				Carácter de la asignatura	
	HCS	HPS	TH	C	HTCS	TH	C	TC	(X) Obligatoria	() Optativa
General	1	5	6	6	0	0	0	6		

SERIACIÓN		
Explícita		Implícita
Asignaturas antecedentes	Asignaturas subsecuentes	Conocimientos previos
Algoritmos	Programación II	

PROPÓSITO DE LA ASIGNATURA	
Desarrollar soluciones computacionales utilizando lenguajes de programación de alto nivel para resolver problemas.	
COMPETENCIAS A DESARROLLAR	
Genéricas	Específicas
Capacidad de análisis y síntesis. Resolución de problemas. Capacidad de aplicar los conocimientos en la práctica. Trabajo autónomo.	Desarrollar sistemas de software integrando tecnologías para la solución de problemas, automatizando los procesos operativos, flujo de información y toma de decisiones en las organizaciones con un enfoque sistémico bajo estándares internacionales.



UNIDAD No. 1	Elementos básicos del lenguaje	Horas estimadas para cada unidad
		12
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
1.1. Conceptos generales. 1.2. Estructura de un programa. 1.3 Datos, tipos de datos. 1.4 Identificadores, constantes y variables. 1.5 Operadores aritméticos, relacionales, lógicos y de asignación. 1.6 Expresiones. 1.6.1 Expresiones aritméticas. 1.6.2 Reglas de prioridad. 1.6.3 Expresiones lógicas. 1.7 Entornos de desarrollo integrados para el lenguaje de programación (IDE).	Evalúa expresiones que impliquen el uso combinado de operadores aritméticos lógicos y relacionales. Codifica en un lenguaje de programación las soluciones a problemas planteados en lenguaje natural y/o fórmulas matemáticas.	Expresiones evaluadas. Código de la solución de un problema.

UNIDAD No. 2	Estructuras secuenciales	Horas estimadas para cada unidad
		10
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
2.1. Instrucciones de entrada y salida. 2.2. Funciones básicas predefinidas.	Resuelve problemas codificando programas en lenguajes de alto nivel empleando estructuras secuenciales.	Código en un lenguaje de programación de alto nivel aplicando estructuras secuenciales.



UNIDAD No. 3	Estructuras selectivas	Horas estimadas para cada unidad	
		12	
CONTENIDOS			
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
3.1. Conceptos de estructuras selectivas. 3.2. Estructura selectiva simple (If-Then) 3.3. Estructura selectiva doble (If-Then-Else) 3.4. Estructuras selectivas anidadas 3.5. Estructura selectiva múltiple (Switch-Case)	Resuelve problemas codificando programas en lenguajes de alto nivel empleando estructuras selectivas.	Código en un lenguaje de programación de alto nivel aplicando estructuras selectivas.	

UNIDAD No. 4	Estructuras repetitivas	Horas estimadas para cada unidad	
		12	
CONTENIDOS			
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
4.1. Conceptos de estructuras repetitivas. 4.2. Estructura repetitiva mientras (While). 4.3. Estructura repetitiva hacer-mientras (Do-While). 4.4. Estructura desde /para (For).	Resuelve problemas codificando programas en lenguajes de alto nivel empleando estructuras repetitivas.	Código en un lenguaje de programación de alto nivel aplicando estructuras repetitivas.	

UNIDAD No. 5	Funciones y modularización	Horas estimadas para cada unidad	
		10	
CONTENIDOS			
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje	
5.1. Conceptos sobre funciones y modularización. 5.2. Funciones definidas por el programador. 5.3. Legibilidad y reusabilidad de código. 5.4. Alcance de las variables. 5.5. Argumentos por valor y referencia.	Crea funciones para ser usadas dentro del código de un lenguaje de programación de alto nivel. Emplea argumentos por valor y por referencia, en la solución de un problema dentro del código de lenguaje de programación.	Funciones definidas por el programador. Código en un lenguaje de programación de alto nivel que utilice funciones definidas por el programador.	



UNIDAD No. 6	Manejo de cadenas	Horas estimadas para cada unidad
		8
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
6.1. Carácter y cadenas. 6.2. Definición de cadenas de caracteres 6.3. Funciones de biblioteca para manejo de cadenas.	Utiliza las cadenas en la solución de un problema codificado en un lenguaje de programación.	Código en un lenguaje de programación de alto nivel que utilice cadenas.

UNIDAD No. 7	Arreglos	Horas estimadas para cada unidad
		16
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
7.1 Introducción a las estructuras de datos. 7.2 Arreglos unidimensionales: vectores. 7.3 Arreglos mutidimensionales. 7.4 Datos de tipo estructura. 7.5 Arreglos de estructuras.	Resuelve problemas codificando programas en lenguajes de alto nivel empleando arreglos.	Código en un lenguaje de programación de alto nivel que utilice arreglos.

UNIDAD No. 8	Archivos	Horas estimadas para cada unidad
		16
CONTENIDOS		
Conceptuales	Aprendizajes esperados	Evidencias de aprendizaje
8.1. Introducción. 8.2. Operaciones primitivas sobre archivos. 8.3. Archivos binarios. 8.4. Archivos de registros. 8.5. Archivos de texto. 8.6. Operaciones lógicas sobre archivos.	Resuelve problemas codificando programas en lenguajes de alto nivel empleando archivos.	Código en un lenguaje de programación de alto nivel que utilice archivos.



Contenidos procedimentales	Contenidos actitudinales
Identifica los tipos de datos. Analiza los requerimientos. Diseña la solución. Codifica la solución. Compilación del programa. Implementación en un entorno de desarrollo integrado.	Responsabilidad en la entrega de sus trabajos. Disposición para trabajar en equipo. Rigor en el manejo de la información. Respeto a las propuestas de sus compañeros.

Metodología para la construcción del conocimiento	
Actividades de aprendizaje con el docente	Actividades de aprendizaje autónomo
Exposición de temas. Prácticas guiadas en el laboratorio de cómputo Prácticas supervisadas. Uso de un entorno de desarrollo integrado.	Desarrollo de programas. Investigación documental y en línea. Uso y análisis de software para el desarrollo de programas. Reproducción de video tutoriales de lenguajes de programación y entornos para desarrollo integrado. Solución de problemas. Prácticas autónomas.

Evidencias de desempeño		
Acreditación	Evaluación	Calificación
Cumplir con lo establecido en el Reglamento Escolar vigente. Entrega de evidencias de aprendizaje.	Al final de cada unidad.	5% Expresiones evaluadas. 5% Códigos de la solución de un problema. 10% Códigos en un lenguaje de programación de alto nivel aplicando estructuras secuenciales. 10% Códigos en un lenguaje de programación de alto nivel aplicando estructuras selectivas. 10% Códigos en un lenguaje de programación de alto nivel aplicando estructuras repetitivas. 5% Funciones definidas por el programador. 10% Códigos en un lenguaje de programación de alto nivel que utilice funciones definidas por el programador. 10% Códigos en un lenguaje de programación de alto nivel que utilice cadenas. 15% Códigos en un lenguaje de programación de alto nivel que utilice arreglos. 20% Códigos en un lenguaje de programación de alto nivel que utilice archivos.



FUENTES DE APOYO Y CONSULTA

BÁSICA

1. Deitel, P. y Deitel, H. (2010). C How to program (6a ed.). New Jersey: Pearson Education. *
2. Joyanes, L. (2013). Fundamentos generales de programación. México: McGraw-Hill.
3. Joyanes, L. (2012). Fundamentos de la Programación. México: McGraw-Hill. *
4. Márquez, G., Osorio, S. y Olvera, N. (2011). Introducción a la programación estructurada en C. México: Pearson educación. *
5. Sznajdleder, P. (2012). Algoritmos a fondo con implementaciones en C y Java. México: Alfaomega. *
6. Méndez, Al. (2013). Diseño de algoritmos y su programación en C. México: Alfaomega.

COMPLEMENTARIAS

1. Bailey, T. (2005). An introduction to the C programming language and software desing. Recuperado de <http://www-personal.acfr.usyd.edu.au/tbailey/ctext/ctext.pdf> *
2. Gottfried, B. (2005). Programación en C. México: McGraw-Hill. *
3. García, F. (2001.). Introducción al lenguaje de programación C. Madrid: Universidad Carlos III, Arcos. *
4. Kernighan, B. W., Ritchie, D. M. (2012). The C Programming Language. Massachusetts, EUA: Prentice-Hall *
5. Marzal, A. y Gracia, I. (2003). Introducción a la programación con C. Castellón, España: Edición Internet Departamento de Lenguajes y Sistemas informáticos, Universitat Jaume I. Recuperado de <http://campuscurico.utalca.cl/~rgarrido/recursos/c.pdf> *

*La bibliografía con antigüedad mayor de cinco años contiene información relevante para el desarrollo de esta asignatura. Cabe destacar que son textos clásicos con ejemplos didácticos de fácil comprensión para el estudiante. Son difíciles de conseguir en el mercado, pero se encuentran en los catálogos de varias bibliotecas.

RESPONSABLE DEL DISEÑO

Elaborado por	Rubén Jerónimo Yedra, Erika Yunuen Morales Mateos.
Fecha de elaboración	20 de diciembre de 2016.