

CARTA AL ESTUDIANTE

CODIGO: EIF 201
NOMBRE: Programación I
REQUISITOS: EIF-200: Fundamentos de Informática.
MAT-030: Matemática para Informática I.
NATURALEZA: teórico/práctico
AREA DISCIPLINARIA: Ingeniería de software
NIVEL: I nivel
CICLO LECTIVO: II ciclo 2017
PROFESORES: Santiago Caamaño Polini, Karol Leitón Arrieta, Marianela Solano Orias, Miguel Arturo Corrales Ureña, José Pablo Calvo Suárez. Jennifer Fuentes Bustos y Karol Leitón Arrieta (coordinadoras)

Créditos	Horas Semanales	Horas Presenciales *		Horas estudio indepen. **
4	10	Prácticas	Teóricas	6
		2	2	

* **Horas Presenciales:** tiempo que el estudiante está en el aula, supervisado por el profesor.

** **Horas estudio independiente:** tiempo que el estudiante de manera independiente se dedica a repasar el material, hacer tareas, ejercicios extras, etc.

DESCRIPCION

Este curso le da continuidad al curso EIF 200 Fundamentos de Informática, en la enseñanza y aprendizaje de los aspectos básicos del diseño orientado a objetos y su respectiva programación, particularmente usando el lenguaje C++. En este curso el estudiante aplicará técnicas de desarrollo de software.

De esta forma el curso de Programación I profundiza la temática desarrollada en el curso anterior, principalmente introduciendo las relaciones entre clases y objetos, colecciones y persistencia de los datos, los cuales le dan al estudiante una perspectiva más completa de la programación de aplicaciones en el paradigma de orientación a objetos, estableciendo las bases del curso siguiente.

OBJETIVO GENERAL

Diseñar e implementar a nivel básico con base en el paradigma de programación orientada a objetos, las principales técnicas de programación y estructuras de datos que permiten al estudiante desarrollar las habilidades necesarias para el desarrollo de software de pequeña a mediana complejidad.

OBJETIVOS ESPECÍFICOS: que el estudiante al finalizar el curso sea capaz de:

1. Identificar y clasificar los conceptos más importantes mediante el paradigma de orientación a objetos para definir la arquitectura de un programa.
2. Analizar y proponer soluciones computacionales a problemas mediante un proceso de descomposición y especialización.
3. Diseñar e implementar soluciones computacionales a problemas a partir del paradigma de orientación a objetos en lenguaje C++.
4. Analizar y discernir adecuadamente las estructuras de programación que permiten resolver un problema, disponibles en el lenguaje de programación C++.
5. Indagar de forma autodidacta temas relacionados con el curso.
6. Propiciar el desarrollo de habilidades para el trabajo en equipo: liderazgo efectivo, colaboración, división efectiva y adecuada de tareas, compromiso con el propio aprendizaje, resolución de conflictos, etc.

ASPECTOS COGNITIVOS

- Identificar y clasificar conceptos a través del paradigma de orientación a objetos para definir la arquitectura de un programa que da solución a un problema.

HABILIDADES DISCIPLINARIAS

1. Analizar problemas y plantear su solución mediante un proceso de descomposición y especialización.
2. Analizar y discernir adecuadamente las estructuras disponibles en el lenguaje de programación C++ para la solución de problemas.
3. Diseñar soluciones a problemas usando el paradigma de POO.
4. Implementar aplicaciones básicas en el paradigma de orientación a objetos en lenguaje C++.

HABILIDADES BLANDAS (*SOFT SKILLS*):

1. Investigar de manera autodidacta acerca de temas relacionados con el curso.
2. Trabajar adecuadamente en equipos de trabajo, asumiendo sus roles y responsabilidades y resolviendo adecuadamente los conflictos.
3. Asumir la responsabilidad de su propio aprendizaje.
4. Leer y asimilar documentos en idioma inglés y español.
5. Elaborar informes acerca de los aprendizajes alcanzados.
6. Evaluar su proceso de aprendizaje y el de sus compañeros de grupo.

CONTENIDOS

Como contenidos lista de temas que deben ser cubiertos en el curso, sin embargo es importante entender que al estudiar el paradigma de programación orientada a objetos esta lista de temas no es un listado estrictamente secuencial.

1. Conceptos básicos de programación orientada a objetos.

- 1.1. Clase, objeto, estado, identidad, atributos, métodos, visibilidad, vista, abstracción, encapsulación.
- 1.2. Acceso directo e indirecto a objetos.
- 1.3. Asignación y liberación de memoria, uso de punteros.

2. Colecciones básicas

- 2.1. Arreglos de tipos básicos y de objetos (dinámicos y estáticos), administración de operaciones básicas.
 - 2.1.1. Unidimensional
 - 2.1.2. Bidimensional
- 2.2. Lista simplemente enlazada, administración de operaciones básicas.
- 2.3. Almacenamiento de colecciones en archivos de texto (persistencia de los datos).

3. Relaciones. (Diseño e implementación)

- 3.1. Principios básicos de la POO: Principio abierto-cerrado. Principio de responsabilidad única. Ley de Deméter.
- 3.2. Bajo acoplamiento y alta cohesión.
- 3.3. Análisis y diseño de soluciones a problemas.
 - 3.3.1. Diseño y conceptualización de clases.
 - 3.3.2. Identificación de clases
 - 3.3.3. Relaciones entre clases
 - 3.3.4. Diagramas de clases
- 3.4. Representación e implementación de cada una de las relaciones entre clases: dependencia, herencia, asociación, agregación y composición, su cardinalidad y navegabilidad.
- 3.5. Herencia (Especialización y Generalización).
 - 3.5.1. Invocación de constructores y destructores
 - 3.5.2. Enlace dinámico
- 3.6. Concepto de polimorfismo.
 - 3.6.1. Sobrecarga de métodos para extensión y sustitución (*overwriting*).
 - 3.6.2. Sentencia virtual, métodos virtuales puros, clases abstractas.

METODOLOGÍA

Para el logro de los objetivos se propone un abordaje de los contenidos a partir de la participación activa del estudiante como persona comprometida con su propio aprendizaje, aprovechando las experiencias vividas, es por eso que en el curso se utilizarán las siguientes estrategias.

- Los temas se abordarán mediante clases magistrales, utilizando ejemplos para apoyar el trabajo de clase.
- Se desarrollarán ejercicios prácticos asociados con los temas.
- Se realizarán laboratorios semanales que le permitan al estudiante practicar los conceptos teóricos guiado por el profesor y continuar con el proceso en su tiempo de estudio independiente.
- Se realizarán diversas actividades como quices, tareas, laboratorios, trabajo en clase y proyectos programados con el fin de promover la asimilación de los conceptos.
- Se promoverá la investigación constructiva en los estudiantes, de tal manera que estos brinden aportes al aprendizaje de los demás y construyan sus propias soluciones a los problemas planteados.
- Cada profesor hará uso de la misma aula virtual como recuso tecnológico de apoyo al curso para el intercambio de materiales, entrega de trabajos, comunicación y participación en actividades.
- Cada profesor seleccionará cual es el ambiente de desarrollo (IDE) a utilizar durante el curso.

Papel del Estudiante:

Mediante las actividades realizadas en el aula y como parte de las horas de estudio independiente se promueve que el estudiante sea una persona altamente participativa, proactiva, independiente y propositiva en el cumplimiento de sus responsabilidades académicas en el curso y que muestre, en todas sus acciones académicas, una conducta ética, en acuerdo con las normas establecidas por la Universidad Nacional.

En este curso se requiere que el estudiante dedique al menos 6 horas de estudio extra clase, asista a las horas de consulta del profesor. La asistencia a clases es la obligatoria, con tres ausencias injustificadas el estudiante reprueba el curso.

Horas de Consulta	
Profesores	Día y hora
Santiago Caamaño Polini	Lunes y Jueves 16:00
Marianela Solano Orias	Lunes y Jueves 11:50
Karol Leitón Arrieta	Viernes 16:00
José Pablo Calvo Suárez	Miércoles 17:00
Miguel Arturo Corrales Ureña	Lunes 13:00

Papel de la cátedra:

La información sobre exámenes, asignación de aulas o cualquier comunicación importante se hará a través del aula virtual institucional. Es responsabilidad del estudiante revisar regularmente el aula virtual para mantenerse informado de las actividades del curso. Por todas estas razones se recomienda desde la coordinación de la carrera, que todos los estudiantes hagan uso de la misma. Para que el estudiante pueda ingresar al aula virtual debe acceder a la página de la Universidad Nacional (<http://www.una.ac.cr>) a la opción Aula Virtual Institucional.

EVALUACIÓN

Cada profesor puede evaluar los laboratorios, trabajos que realiza cada estudiante en horas de clase.

Descripción	Porcentaje
Proyecto(s) (2) Se realizarán entregas en pareja, que servirán para evaluar aspectos prácticos concretos de los temas estudiados en el curso. La formación de las parejas la realiza el profesor.	30%
Primer examen Habilidades disciplinarias: 1,2,3	25%
Segundo examen Habilidades disciplinarias: 1,2,3,4	30%
Actividades que promueven aprendizaje Tareas, Laboratorios y/o pruebas cortas (quices)	15%
Total	100%

Por la naturaleza de los contenidos del curso, es inevitable que los contenidos desarrollados sean acumulativos, es decir, aunque en una evaluación se deba profundizar en uno o más temas específicos, esto no implica que no se puedan incluir temas anteriormente evaluados.

Al ser un curso cuya evaluación contempla aspectos que se desarrollan a lo largo del curso, como los laboratorios y proyectos programados no tiene examen extraordinario, por lo que la suma de los porcentajes obtenidos por el estudiante en los rubros anteriores determina su nota de aprovechamiento (NA), si esta es superior o igual a 70% el estudiante aprueba el curso, y si la NA es menor a 70% el estudiante reprueba el curso.

CRONOGRAMA

FECHA ESTIMADA	ACTIVIDAD
SEMANA 1 (24 de julio al 29 de julio)	Inicio de Lecciones <i>Feriado: 25 de Julio de 2017. Día de la Anexión de Guanacaste.</i>
SEMANA 2 (31 de julio al 05 de agosto)	
SEMANA 3 (07 de agosto al 12 de agosto)	<i>Feriado: 02 de Agosto de 2017. Día de la Virgen de los Ángeles.</i>
SEMANA 4 (14 de agosto al 19 de agosto)	Entrega enunciado Proyecto#1 <i>Feriado: 15 de Agosto de 2017. Día de la Madre.</i>
SEMANA 5 (21 de agosto al 26 de agosto)	
SEMANA 6 (28 de agosto al 02 de setiembre)	
SEMANA 7 (04 de setiembre al 09 de setiembre)	
SEMANA 8 (11 de setiembre al 16 de setiembre)	Entrega solución Proyecto#1 <i>Feriado: 15 de Setiembre de 2017. Día de la Independencia de Costa Rica.</i>
SEMANA 9 (18 de setiembre al 23 de setiembre)	1er Examen (Domingo 24 de setiembre, 2017)
SEMANA 10 (25 de setiembre al 30 de setiembre)	Entrega enunciado Proyecto#2
SEMANA 11 (02 de octubre al 07 de octubre)	
SEMANA 12 (09 de octubre al 14 de octubre)	
SEMANA 13 (16 de octubre al 21 de octubre)	<i>Feriado: 16 de Octubre de 2017. Celebración del Día de las Culturas.</i>
SEMANA 14 (23 de octubre al 28 de octubre)	
SEMANA 15 (30 de octubre al 04 de noviembre)	Entrega solución Proyecto#2
SEMANA 16 (06 de noviembre al 11 de noviembre)	Finalizan lecciones, 2do Examen (Domingo 12 de noviembre, 2017)

BIBLIOGRAFIA

1. Deitel, Harvey y Deitel Paul. (2009). *Como programar en C++*. Prentice Hall. 6ta Edición. México.
2. Joyanes, Luis. (1996). *Programación Orientada a objetos: conceptos, modelado, diseño y codificación en C++*. McGraw Hill. 1 Edición, España.
3. Langsam Yedidiah, Augenstein Moshe, Tenenbaum Aaron. (1997). *Estructuras de Datos con C y C++*. Pearson Education, 2da Edición. USA.
4. Rodríguez Rojas, Oldemar; (1997). “ *C ++ para ambientes gráficos*”. Editorial Tecnológica de Costa Rica, 1era edición, Costa Rica
5. Savitech Walter. (2000) *Resolución de problemas con C++*. Prentice Hall, 1 Edición.
6. Schildt, Herbert. (1995). *C++ Guía de Auto enseñanza*. Osborne/McGraw-Hill, Primera edición.
7. Sedgewick, Robert. (1995). *Algoritmos en C++* .Prentice Hall. 1 Edición.
8. Van Gelder, Baase. (2002). *Algoritmos computacionales*. Prentice Hall. 3era Edición.

ESPECIFICACIONES GENERALES

1. Es requisito indispensable para ganar el curso la presentación de todos los proyectos.
2. Cada proyecto debe ser entregado según las indicaciones de cada profesor. La documentación que el estudiante debe entregar será claramente establecida por la cátedra de profesores.
3. No se reciben trabajos luego de la fecha y hora de entrega indicadas por el profesor del curso.
4. El mismo día de presentación de los entregables (o en una fecha a convenir con el profesor de curso) se debe realizar una defensa oral del proyecto por parte de todo el grupo.
5. Los exámenes se realizarán de manera coordinada por todos los profesores del curso.
6. En caso de corroborarse algún fraude en la aplicación de alguna evaluación escrita o en la documentación, algoritmos o implementación de las tareas o proyectos, la Escuela de Informática aplicará las sanciones establecidas en el reglamento interno de la Universidad Nacional, y se debe levantar un acta del mismo que será resguardada por la Institución.
7. El horario disponible para la atención a estudiantes será programado y comunicado por cada profesor, la asistencia oportuna y comprometida del estudiante le permitirá obtener del profesor en este espacio: orientación en trabajos asignados durante todo el curso, evacuación de dudas de temas abordados y la articulación conjunta de ideas para el desarrollo de los trabajos.