# 

**Universidad Autónoma de Querétaro**

**Facultad de Informática**

Programación Orientada a Objetos

Plan de Estudios - 2018

**Información Curricular**

|  |  |  |  |
| --- | --- | --- | --- |
| **I. Datos generales de la asignatura** | | | |
| Nombre de la asignatura | Programación Orientada a Objetos | | |
| Clave | 2057 | Número de créditos | 5 |
| Horas prácticas | 2 | Horas teóricas | 2 |
| Programas donde se imparten | SOF18, LAT18, INF18, TEL18. | Semestre | 2 |
| Ubicación de la materia en el mapa curricular | La materia se ubica en el segundo bloque, es la continuación de la materia “Introducción a la Programación”. | | |

|  |
| --- |
| **II. Presentación** |
| La Programación Orientada a Objetos (POO) es una forma de programar específica, más cercana a cómo expresamos las cosas en la vida real, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.  La base de la POO en el **concepto de clases y objetos**. En este estilo de programación se crean aplicaciones a partir de piezas simples y reutilizables que son especificadas a través de clases, que al ser instanciadas se convierten en objetos. |
| **III. Justificación** |
| La programación orientada a objetos busca crear programas de calidad, en los que sus componentes sean extensibles, reutilizables y sea fácil darles mantenimiento.  En la programación orientada a objetos, los datos de entrada son manipulados por los objetos para entregar una salida específica, y cada objeto ofrece una funcionalidad especial. El resultado final de la POO es la creación de aplicaciones o de componentes de software reutilizables. |
| **IV. Objetivo General** |
| El estudiante aprenderá a analizar y desarrollar aplicaciones usando el paradigma orientado a objetos y sus principales características que son abstracción, encapsulación, herencia y polimorfismo. |
| **V. Metodología de enseñanza-aprendizaje** |
| * Enfoque en competencias * Aprendizaje Basado en Problemas * Aprendizaje Basado en la Investigación * Centrar los procesos educativos en el aprendizaje. * Favorecer aprendizajes significativos en los alumnos. * Promover la construcción del aprendizaje a través de la Teoría Sociocultural de Vygotsky. * Trabajo interdisciplinario y multidisciplinario, para generar en el alumno aprendizajes significativos.   Trabajo colaborativo que fomente la responsabilidad, la tolerancia, el respeto, la creatividad y originalidad, además de propiciar la reflexión y la crítica. |
| **VI. Contenido temático** |
| **Unidad 1.** Introducción al paradigma orientado a objetos  **Competencias a desarrollar.**  Comprender los conceptos principales del paradigma de programación orientado a objetos y aplicarlos usando la representación gráfica que aporta UML.  **Objetivo de la unidad.**  Entender el paradigma orientado a objetos y cómo modelar usando UML  **Temas**  1.1Paradigmas de programación  1.1.1 Evolución de la programación estructurada  1.1.2 Conceptualización de la POO  1.2 Introducción a la Programación Orientada a Objetos  1.2.1 Concepto de objeto y sus relaciones  1.3 Representación gráfica de la POO (UML)  1.4 Principios básicos de la POO |
| **Unidad 2.** Programación Orientada a Objetos  **Competencias a desarrollar.**  Implementar clases completas usando atributos, métodos y constructores.  Implementar sobrecarga de métodos para optimizar el código de una clase.  Implementar instancias de objetos, así como usar de forma correcta distintas colecciones de objetos.  **Objetivo de la unidad.**  Entender e implementar programas que utilicen las clases y colecciones.  **Temas**  2.1 Clases  2.1.1 Atributos  2.1.2 Métodos  2.1.3 Modificadores de acceso  2.1.5 Constructores  2.2. Objetos  2.2.1 Instancias  2.2.2 Parámetros  2.3. Colecciones  2.3.1 Creación colecciones de objetos  2.3.2 Manejo de colecciones de objetos |
| **Unidad 3.** Propiedades de la Programación Orientada a Objetos  **Competencias a desarrollar.**  Implementar herencia a través de clases derivadas que permitan reutilizar y sobrescribir a los miembros de una clase.  Implementar clases abstractas, clases estáticas e interfaces para crear jerarquías y colaboraciones entre clases.  **Objetivo de la unidad.**  Aplicar los conceptos de la programación orientada a objetos que permitan construir jerarquías de clases  **Temas.**  3.1 Herencia  3.2 Polimorfismo  3.3 Sobrecarga de operadores  3.4 Clases abstractas  3.5 Clases estáticas  3.6 Interfaces |
| **VII. Formas de evaluación** |
| **Evaluación diagnóstica**: su propósito es establecer un vínculo significativo entre lo que el estudiante sabe, piensa o siente antes de iniciar su proceso enseñanza-aprendizaje sobre el contenido a abordar.  **Evaluación formativa:** ocurre durante todo el proceso enseñanza-aprendizaje y juega un importante papel regulador en dicho proceso, ya que permite conocer los aprendizajes logrados y retroalimentar a los estudiantes y al docente.  **Evaluación sumativa**: modalidad que implica recuperar todas las actividades que permiten dar cuenta del avance individual de los alumnos, al final de cada tema y al término del curso.  **Independientes:** Definir los proyectos y experiencias de aprendizaje que realizarán los estudiantes, así como las rúbricas o instrumentos de evaluación que se usarán.  **Instrumentos de evaluación:**   * Prácticas en clase * Tareas * Proyecto final   **Parciales**   |  |  |  |  | | --- | --- | --- | --- | | Parcial | Unidades | Mes | Periodo | | 1° | 1ra. | Marzo | Enero-Mayo | | 2° | 2da. | Abril | Enero-Mayo | | 3° | 3era. | Mayo | Enero-Mayo | | Final | 1ra, 2da y 3era | Julio | Enero-Mayo |  |  |  |  |  | | --- | --- | --- | --- | | Parcial | Unidades | Mes | Periodo | | 1° | 1ra. | Septiembre | Agosto-Diciembre | | 2° | 2da. | Octubre | Agosto-Diciembre | | 3° | 3era. | Noviembre | Agosto-Diciembre | | Final | 1ra, 2da y 3era | Diciembre | Agosto-Diciembre |   **Evaluación Parcial**   * Asistencia--------------- Obligatoria al 65% * Tarea del Parcial ----- Derecho a Examen * Parciales 70%   + Prácticas --------------- 40%   + Exámenes ------------- 60% * Proyecto Final 30%   **Calificaciones**  0 – 59 = NA  60 - 79 = Examen final  80-100 = Exentos |
| **Bibliografía** |
| Romano, F., Baka, B., & Phillips, D. (2019). Getting Started with Python. Packt Publishing  López, B. (2016). Curso de programación orientada a objetos en C#. México  Sznajdleder, P. (2016). *Java a fondo curso de programación*. México: Alfaomega.  López, L. (2013). Metodología de la Programación Orientada a Objetos. México  Joyanes, L. (2011). Programación en Java 6. Mc Graw Hill  Dietel H., Dietel P., (2008 Séptima Edición). Como programar en Java, Pearson Education, ISBN 978-970-26-1190-5  Barker, J. (2005 Segunda Edición) Beginning Java Objects: From Concepts to Code, Apress  López, L. (2006). Programación estructurada y orientada a objetos. México  Metsker, S. J. (2002). Design Patterns Java Workbook. Addison-Wesley.  Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2003). Patrones de diseño (1.a ed.). Pearson Education |