

Campus: São José dos Campos		
Curso (s): BCC/EC/BCT		
Unidade Curricular (UC): Programação Orientada a Objetos		
Unidade Curricular (UC): <i>[nome da UC em inglês] Object-Oriented Programming</i>		
Unidade Curricular (UC): <i>[nome da UC em espanhol - opcional]</i>		
Código da UC:		
Docente Responsável/Departamento: Otávio Augusto Lazzarini Lemos		Contato (e-mail): otavio.lemos@unifesp.br
Docente (s) Colaborador/a (es/as)/Departamento (s):		Contato (e-mail): [opcional]
Ano letivo: 2022	Termo: 2	Turno/Turma: IA/NA
Nome do Grupo/Módulo/Eixo da UC (se houver):		Idioma predominante em que a UC será oferecida: <input checked="" type="checkbox"/> Português <input type="checkbox"/> English <input type="checkbox"/> Español <input type="checkbox"/> Français <input type="checkbox"/> Libras <input type="checkbox"/> Outro:
UC: <input checked="" type="checkbox"/> Fixa <input type="checkbox"/> Eletiva <input type="checkbox"/> Optativa	Oferecida como: <input checked="" type="checkbox"/> Disciplina <input type="checkbox"/> Módulo <input type="checkbox"/> Estágio <input type="checkbox"/> Outro:	Oferta da UC: <input checked="" type="checkbox"/> Semestral <input type="checkbox"/> Anual
Ambiente Virtual de Aprendizagem: <input type="checkbox"/> Moodle <input checked="" type="checkbox"/> Classroom <input type="checkbox"/> Outro: <input type="checkbox"/> Não se aplica		
Pré-Requisito (s) - Indicar Código e Nome (s) da (s) UC:		
Carga horária total (em horas): 72h		
Carga horária teórica (em horas): 36h	Carga horária prática (em horas): 36h	Carga horária de extensão (em horas, se houver):
Se houver atividades de extensão, indicar código e nome do projeto ou programa vinculado na Pró-Reitoria de Extensão e Cultura (ProEC):		
Ementa: Introdução à Programação Orientada a Objetos; Classes e Métodos; Encapsulamento e Sobrecarga; Sobreposição de Métodos; Construtores e Destrutores; Herança; Polimorfismo e Ligação Dinâmica; Introdução a uma linguagem OO (Tipos de Dados, Operadores, Variáveis, Arrays, Controle de Fluxo); Tratamento de Exceção; Manipulação de Strings e Arquivos; Aplicações; Estudos de Caso.		
Conteúdo programático:  1. Introdução à programação orientada a objetos 2. Classes, Métodos e Atributos 3. Construtores e sobrecarga 4. Atributos e métodos estáticos 5. Estruturas de controle e decisão 6. Reutilização de classes (Herança) 7. Classes abstratas e interfaces 8. Pacotes de classes 9. Arrays e Matrizes 10. Classes de manipulação de strings e arquivos 11. Coleções de objetos 12. Tratamento de Exceção		
Objetivos:		

Gerais: Capacitar o aluno para o desenvolvimento de software orientado a objetos, utilizando uma linguagem de programação com grande aceitação no meio comercial e acadêmico.

Específicos: Propiciar ao aluno uma adaptação (transição) entre a programação estruturada/procedimental e a programação orientada a objetos; Projetar, implementar, testar e depurar programas orientados a objetos; Introduzir os conceitos de classes e objetos, herança e polimorfismo; e Apresentar uma visão geral dos recursos avançados da linguagem.

Metodologia de ensino:

A disciplina será intercalada por aulas teóricas e aulas práticas em laboratório. Nas aulas teóricas serão apresentados os principais conceitos e seus relacionamentos. Já nas aulas de laboratório, os conceitos serão implementados principalmente em linguagem Python e/ou TypeScript, utilizando-se ferramentas de codificação e testes baseados em software livre. Ademais, desenvolver-se-á atividades à distância, com o apoio da ferramenta Classroom.

A metodologia de ensino baseada na resolução de problemas (Problem Based Learning) será amplamente utilizada. O professor, após apresentar a teoria necessária, irá propor problemas e atuará apenas como facilitador junto aos alunos na resolução do problema.

Avaliação: Serão aplicadas duas provas escritas, cuja média formará 80% da nota final. Serão propostos exercícios práticos e teóricos, cuja média formará 20% da nota final.

Bibliografia:

Básica:

Steven F Lott, Dusty Phillips. Python Object-Oriented Programming - Fourth Edition: Build robust and maintainable object-oriented Python applications and libraries.

Boris Cherny. Programming Typescript: Making Your JavaScript Applications Scale.

Sandi Metz. 99 Bottles of OOP - 2nd Edition

Complementar:

Videoaulas, artigos científicos e demais materiais complementares serão disponibilizados na página do curso.

Cronograma: *[opcional]*