



Unit of Study Information

Code	Unit	Evaluation Method	Mode	Session options
CSE20	Programming Techniques	Grade and Attendance	Presencial	Semestral

Workload					
TC	PC	OA	SPA	PACC	Total
2	2	4	8	0	60
<ul style="list-style-type: none">• TC: Theorethic Classes (per week);• PC: Practical Classes (per week);• OA: Out-of-class Activities (hours per session);• SPA: Supervised Practical Activities (classes per session);• PACC: Practical Activities as Curricular Components (classes per session, included in OA and SPA);• Total: total workload in hours.					

Learning Outcomes		
Introduce elementary principles of various computer programming paradigms. Provide the student with in-depth knowledge about the object-oriented paradigm, emphasizing the use of OO mechanisms in conjunction with design patterns. Develop the thinking on how to approach computational problems using this knowledge and practice their application through the use of an OO-supported programming language, an integrated development environment, and component libraries.		
Syllabus		
Introduction to programming language paradigms (declarative: functional and logical; imperative: unstructured, structured, object oriented and concurrent). Study of the Object Oriented Programming Paradigm and associated basic representations. Practical and project implementation activities in object oriented paradigm.		
Content		
Order	Syllabus	Content
1	Introduction to programming language paradigms (declarative: functional and logical; imperative: unstructured, structured, object oriented and concurrent)	Basic concepts of functional programming and logical programming. Imperative concurrent programming basics: event orientation. Introduction to imperative object oriented programming.
2	Study of the Object Oriented Programming Paradigm and associated basic representations.	Abstraction and encapsulation, classes, objects, attributes and methods. Object lifecycle management: constructors and destructors, object reference, object relationships, persistence. Inheritance and multiple inheritance. Class properties, attributes, and methods: visibility and scope. Polymorphism. Overload of methods and operators, substitution principle and dynamic linking. Abstract classes. Nested classes. Templates. Introduction to Using UML Class Diagrams
3	Practical and project implementation activities in object oriented paradigm.	Introduction to object-oriented design patterns. Application of object orientation in software design.

Basic Resources
DEITEL, Harvey M.; DEITEL, Paul J. (Autor). Java, como programar. 6. ed. São Paulo, SP: Pearson Prentice Hall, 2005. xl, 1110 p. + 1 CD-ROM (4¾ pol.) ISBN 8576050196.
GAMMA, Erich et al. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre, RS: Bookman, 2000 xii,364 p. ISBN 8573076100.
DEITEL, Harvey M.; DEITEL, Paul J. C++ como programar. 5. ed. Porto Alegre, RS: Bookman, 2001. xxviii, 1098 p. + CD-ROM ISBN 85-7307-740-9.

Aditonal Resources
STROUSTRUP, Bjarne. The C++ programming language. 3rd ed. Reading, Mass.: Addison-Wesley, 1997. x, 910 p. ISBN 0-201-88954-4.
GOETZ, Brian. Java concurrency in practice. Upper Saddle River, NJ.: Addison-Wesley, c2006. 403 p. ISBN 0321349601
LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre, RS: Bookman, 2007. 695 p. ISBN 85-363-0358-1.
BARNES, David J.; KÖLLING, Michael. Programação orientada a objetos com Java: uma introdução prática usando o BlueJ. São Paulo: Pearson Prentice Hall, 2004. xxviii, 368 p. + 1 CD-ROM ISBN 9788576050124.
LEA, Douglas. Concurrent programming in Java: design principles and patterns. 2nd ed. Reading, Mass.: Addison-Wesley, c2000. 411 p. (Java series) ISBN 0201310090
SIERRA, Kathy; BATES, Bert. Use a cabeça! Java. 2. ed. Rio de Janeiro, RJ: Alta Books, 2007. 470 p. ISBN 9788576081739.
ECKEL, Bruce. Thinking in Java. 4th. ed. Upper Saddle River: Prentice-Hall, 2007. 1482 p. ; ISBN 0-13-027363-5