COURSE SYLLABUS  
**CSC10003 – Objected Oriented Programming**

# GENERAL INFORMATION

|  |  |
| --- | --- |
| Course name: Object-Oriented Programming | |
| Course name (in Vietnamese): Phương pháp Lập trình hướng đối tượng | |
| Course ID: CSC10003 |  |
| Knowledge block: |  |
| Number of credits: | 4 |
| Credit hours for theory: | 45 |
| Credit hours for practice: | 30 |
| Credit hours for self-study: | 90 |
| Prerequisite: |  |
| Prior-course: |  |
| Instructors: Nguyen Van Vu, Tran Duy Quang |  |

# COURSE DESCRIPTION

This course provides students fundamental knowledge of object-oriented programming (OOP). Students will learn core concepts of OOP such as class, object, reuse, constructor, destructor, operator, encapsulation, inheritance, polymorphism, overloading and overriding. They will obtain knowledge and thinking in object-oriented methodology. Students will practice object-oriented programming skills using C++ language and/or Java through individual and group-based projects

# COURSE GOALS

At the end of the course, students are able to

|  |  |  |
| --- | --- | --- |
| **ID** | **Description** | **Program LOs** |
| G1 | Differenciate the difference between procedural programming and OOP |  |
| G2 | Understand and describe core concepts of OOP such as class, object, reuse, constructor, destructor, operator, encapsulation, inheritance, polymorphism, overloading and overriding |  |
| G3 | Model simple real-world problems using UML class-diagrams |  |
| G4 | Apply the understanding of core concepts of OOP to implement programs using C++ |  |
| G5 | Understand the difference between C++ and Java in implementing OOP concepts |  |

# COURSE OUTCOMES

|  |  |  |
| --- | --- | --- |
| **CO** | **Description** | **I/T/U** |
| G1.1 | Able to differenciate the difference between procedural programming and OOP | T |
| G2.1 | Understand core concepts of OOP class, object, reuse, constructor, destructor, operator, encapsulation, inheritance, polymorphism, overloading and overriding | T, U |
| G2.2 | Describe core concepts of OOP class, object, reuse, constructor, destructor, operator, encapsulation, inheritance, polymorphism, overloading and overriding | T, U |
| G3.1 | Understand basic class-related notations of Unified Modeling Language (UML) | T, U |
| G3.2 | Use UML class diagram notations to model real-world problems | T, U |
| G4.1 | Write OOP programs to implement real-world problems using C++ | T, U |
| G5.1 | Describe the difference between C++ and Java in implementing OOP concepts | T |
| G5.2 | Write simple programs using Java | U |

# TEACHING PLAN

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Topic** | **Course outcomes** | **Teaching/Learning Activities** |
| 1 | Introduction to OOP | G1.1, G2.1 | Lecture.  Practice. |
| 2 | UML and class diagram  Constructor and Destructor | G3.1, G3.2  G2.1, G2.2 | Lecture.  Practice. |
| 3 | Operators | G2.1, G2.2, G5.1, G5.2 | Lecture.  Practice. |
| 4 | Law of the Big Three | G2.1, G2.2 | Lecture.  Practice. |
| 5 | Inheritance | G2.1, G2.2 | Lecture.  Practice. |
| 6 | Midterm |  | Exam |
| 7 | Polymorphism | G2.1, G2.2 | Lecture.  Practice. |
| 8 | Exception Handling | G2.1, G2.2 | Lecture.  Practice. |
| 9 | Templates and STL | G2.1, G2.2 | Lecture.  Practice. |
| 10 | Patterns | G2.1, G2.2, G3.2, G3.2 | Seminar.  Practice. |
| 11 | Review and summary | G2.1, G2.2, G5.1, G5.2 | Lecture. |

# ASSESSMENTS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Topic** | **Description** | **Course outcomes** | **Percentage** |
| **Quiz** | Short tests and excercises in theory lectures. | From 3 to 4 quizzes/assignments that are given randomly, without prior notice. This assignment is used to check student attendance and attention. | G2.1, G2.2 | **10%** |
| **LabA** | Lab assignment | Assignments given in lab sessions | G4.1, G2.1 | **10%** |
| **LabP1** | Lab project 1 | A mid-term project given in lab sessions | G4.1, G5.2, G3.2 | **15%** |
| **LabP2** | Lab project 2 | A final project given in lab sessions | G4.1, G5.2 | **15%** |
| **Midterm** | Midterm | Short exam in the middle of the class | G1, G2, G3 | **15%** |
| **Final** | Final exam | Final exam covers all theory lectures and lab exercises | G1, G2, G3, G5 | **35%** |
| **Bonus** | Extra credit | Bonus of up to 10% for optional extra work performed during theory lectures or lab sessions |  | **10%** |

# RESOURCES

* Main Text
  + Object Oriented Programming using C++, Pohl, Addison Wesley.
* Additional Materials
  + C++ FAQ Lite, Marshall Cline, <http://parashift.com/c++-faq-lite/>
  + C++ Primer Plus, 4th Edition, Stephen Prata, SAM, 2001.
  + Effective and More Effective C++, Scott Meyers.
  + The C++ Programming Language, Bjarne Stroustrup, Addison-Wesley.
  + Lập trình hướng đối tượng, Trần Đan Thư, Đinh Bá Tiến, Nguyễn Tấn Trần Minh Khang.

# GENERAL REGULATIONS & POLICIES

* Students are responsible for reading and following strictly the regulations and policies of the Faculty and University.
* Students who are absent for more than 3 theory sessions are not allowed to take the final exam.
* Any kind of cheating and plagiarism is prohibited. Students committing cheating and plagiarizing will receive zero for the course. Incidents are then submitted to the school and university for further review.
* Students are encouraged to form study groups to discuss. However, individual work must be done and submitted by each student individually.