

## Course Syllabus

### Object-Oriented Programming & Problem Solving 7.5 Credits\*, First Cycle

#### Learning Outcomes

##### *Knowledge and Understanding*

After completing the course students shall be able to:

- explain the concept of object orientation
- describe basic terms, concepts and keywords in software development

##### *Skills and Abilities*

After completing the course students shall be able to:

- develop language-independent algorithms aimed at a structured design when developing programs
- reuse finished program libraries in their own program development
- create interactive applications
- use selective statements that contain conditions for controlling the execution of a program
- use iteration statements to repeatedly execute statements until certain conditions are met
- create and use custom classes containing properties and operations that represent the behaviour of objects
- create and use objects as instances of classes
- create and use functions / methods
- use data structures to (temporarily) store and manipulate data during execution
- write down data to files or read data from files
- present (in writing and orally) explain, summarise and discuss selected solution methods and results obtained
- troubleshoot programs
- document programs using comments and using parts of the UML (Unified Modeling Language)

#### Course Content

The course deals with concepts in object oriented programming (OOP) and how object oriented programming languages are generally structured. Further, basic language elements

are treated as identifiers, data types, modifiers, operators, variables, expressions and statements. Particular emphasis is on classes and objects that are used in the building of programs. In addition, the course looks at the process of allocation of variables, constants and object references. The course also deals with how related documentation is written. Furthermore, it deals with how decisions can be made through logical choices (conditionally controlled) during program flow and how so-called functions / methods are created to perform operations on data. Furthermore, various types of iteration are treated, i.e., how statements are repeated until certain pre defined conditions are met. Students learn how to create their own classes and how objects are created by such classes. There is also discussion on how data can be written to and read from files, and how data sets are stored in data structures.

**Assessment**

Test and assignments.

**Forms of Study**

Lectures and exercises.

**Grades**

The Swedish grades U–G.

**Prerequisites**

General entry requirements and Mathematics 2a or 2b or 2c, Social Sciences 1b or 1a1 + 1a2  
No knowledge of Swedish is required.

**Other Information**

The course is equivalent to Introduction to Object-Oriented Programming (IK1052), Introduction to Java Programming (IK1046) and Software Engineering, Introduction (IK1085), and can therefore not be used towards the same degree.

The maximum number of exam re-sits is limited to five per module.

A computer with a Windows operating system and an Internet broadband connection are required for the course. Other operating systems are accepted but not supported. In addition a web camera and a headset, or a microphone, are required.

**Subject:**

Information Systems

**Group of Subjects:**

Informatics/Computer and Systems Sciences

**Disciplinary Domain:**

Technology, 100%

**This course can be included in the following main field(s) of study:**

1. Computer Engineering
2. Information Systems
3. Microdata Analysis

**Progression Indicator within (each) main field of study:**

1. G1N
2. G1N
3. G1N

**Approved:**

Approved 21 February 2019

Valid from 21 February 2019