

# Service-Oriented Programming

---

## *Course Module 1*

Version 1.1  
March 2, 2014

**Xumin Liu, Rajendra Raj**

Computer Science  
RIT  
*[xl, rkr]@cs.rit.edu*

**Tom Reichlmayr**

Software Engineering  
RIT  
*tjr@se.rit.edu*

**Chunmei Liu**

Systems & Computer Science  
Howard University  
*chunmei@scs.howard.edu*

**Alex Pantaleev**

Computer Science  
SUNY at Oswego  
*alex@cs.oswego.edu*

This material is based upon work supported by the National Science Foundation under Awards DUE-1140567, DUE-1141112, and DUE-1141200. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## 1. Overview

This course module is the first of three modules being developed to teach Service-Oriented Programming (SOP) to undergraduate students in computing. Course modules are well-defined, self-contained units of instruction that may be incorporated, either as-is or with minor modifications, within one or more courses offered in diverse disciplines. This modular approach also allows for easy vertical integration of SOP into the undergraduate curriculum in introductory, mid-level and senior courses in disciplines such as Computer Science and Software Engineering.

This course module targets CS2, the second course in the problem solving sequence typically required for first year students in several computing majors such as Computer Science, Software Engineering, Computer Engineering, and Computational Math. The lecture materials provide a brief overview of service-oriented architecture and the fundamental principles of SOP. Topics span the motivation and historical overview of SOP, a look at some typical SOP applications, a comparison of SOP with other problem solving methodologies, and an introduction to the state-of-the-art SOP technologies.

**Prerequisite knowledge.** This course module is typically meant for students who have some knowledge of programming in an object-oriented language, specifically Java or C#, which are major languages used in introductory computing courses across the U.S. and the world.

**Module Learning Outcomes.** Students completing this course module will be able to:

- Describe why service-oriented programming is important in modern programming.
- Explain the basic concepts of web services and service-oriented architecture.
- Develop an application that reuses previously implemented web services.

## 2. Rationale

The global service-oriented middleware market will grow to at least \$8.2 billion by 2016, according to a WinterGreen Report<sup>1</sup>. It is crucial that undergraduate Computer Science (CS) and Software Engineering (SE) students learn about these developments in web services and service-oriented computing as these concepts and technologies will dominate software development for the next few years.

## 3. Recommended use

The authors have typically targeted this module to a course such as CS2, but the module may also be used in any course where the students have the prerequisite knowledge of object-oriented programming.

---

<sup>1</sup> WinterGreen Research, Inc. SOA Application Middleware – Markets Reach \$8.2 Billion by 2016. In <http://www.wintergreenresearch.com/reports/SOA%202010%20press%20release.pdf>, 2010.

## **4. Lecture Slides**

The lecture slides for this module are also included in this folder. The instructor may use them as-is or incorporate them into his or her materials.

## **5. Sample questions**

The sample questions for this module, also included in this folder, are meant for low stakes quizzes. The instructor may use them as-is or incorporate them into his or her tests.

## **6. Labs/programming assignments**

A sample lab assignment is provided in this folder to permit students gain hands-on experience in making use of existing web services in their programming.

## **7. FAQ**

This section will be populated after the authors have gained further experience with the use of these modules in actual courses.

## **8. Readings**

These readings are primarily for faculty teaching these courses; a subset of the material may also be handed out to students as appropriate.

1. Juval Lowy, Programming WCF Services: Mastering WCF and the Azure AppFabric Service Bus, O'Reilly Media; 3rd Edition, 2010.
2. Eben Hewitt, Java SOA Cookbook, O'Reilly Media; 1st Edition, 2009.

## **9. Links**

These links are primarily for faculty teaching these courses; a subset of the material may also be handed out to students as appropriate.

1. <http://msdn.microsoft.com/en-us/library/aa480021.aspx>
2. [https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swapp&S\\_PKG=ov1152&S\\_TACT=109KA8GW&S\\_CMP=web\\_ibm\\_xx\\_soa\\_bd](https://www14.software.ibm.com/webapp/iwm/web/signup.do?source=swapp&S_PKG=ov1152&S_TACT=109KA8GW&S_CMP=web_ibm_xx_soa_bd)
3. <http://www.w3schools.com/webservices/>

## **10. Module Evaluation**

This section will include assessment tools to measure student learning and module effectiveness. In addition to direct student assessment using quizzes, assignments and tests, module evaluation requires student self-assessment and instructor self-assessment of the materials. The latter assessment materials will be provided to instructors who are early adopters of this course module.