东北大学秦皇岛分校本科课程教学大纲

一、课程基本信息

<table>
<tr><td>开课单位</td><td colspan="3">东北大学
悉尼智能科技学院</td><td>课程代码</td><td colspan="5">3100214004</td></tr>
<tr><td>中文名称</td><td colspan="8">软件分析综合实践</td></tr>
<tr><td>英文名称</td><td colspan="8">Software Analysis Studio</td></tr>
<tr><td>总学时</td><td>3W</td><td rowspan="2">理论学时</td><td rowspan="2">3W</td><td rowspan="2">实验学时</td><td rowspan="2">0</td><td rowspan="2">上机学时</td><td rowspan="2">0</td><td rowspan="2">设计学时</td><td rowspan="2">0</td><td rowspan="2">课外学时</td><td rowspan="2">0</td></tr>
<tr><td>学　分</td><td>3</td></tr>
<tr><td>课程类型</td><td colspan="8">☐理论课程　☐理论（含实验）课程　☑实践类课程</td></tr>
<tr><td>课程类别</td><td colspan="8">☐通识类　☐学科基础类　☐专业方向类　☑实践类</td></tr>
<tr><td>适用专业</td><td colspan="8">计算机科学与技术</td></tr>
<tr><td>先修课程</td><td colspan="8">C++程序设计基础</td></tr>
<tr><td>建议开课学期</td><td colspan="8">☐1-1 ☐1-2 ☐2-1 ☑2-2 ☐3-1 ☐3-2
☐4-1 ☐4-2 ☐5-1 ☐5-2 ☐*（代表每学期都开设）</td></tr>
<tr><td>考核方式</td><td colspan="3">☐考试 ☑考查</td><td>成绩分制</td><td colspan="4">☑百分制 ☐五级制 ☐二级制</td></tr>
<tr><td>使用教材</td><td colspan="8">无</td></tr>
<tr><td>参考书及教学资源</td><td colspan="8">1、Static Value-Flow Analysis Framework for Source Code
https://github.com/SVF-tools/SVF
https://github.com/SVF-tools/Teaching-Software-Analysis
2、Compilers: Principles, Techniques, and Tools Hardcover,
https://www.amazon.com.au/Compilers-Alfred-V-Aho/dp/0321486811
3、LLVM Compiler
https://llvm.org/
4、Anders Møller and Michael I. Schwartzbach, Static Program Analysis,
https://cs.au.dk/~amoeller/spa/spa.pdf</td></tr>
<tr><td>其他说明</td><td colspan="8">无</td></tr>
</table>

二、课程简介

中文描述

本课程是计算机科学与技术专业开设的实践类课程。软件系统深深植根于广泛的行业和企业中，非常普遍，以至于在软件缺陷出现之前，我们往往意识不到它的存在。一个 bug 就可能导致严重的软件故障，从而造成巨大的社会和经济影响。尽管在提高软件可靠性和安全性方面越来越受到重视，但现代复杂系统（例如，包含数百万行代码）仍然存在漏洞。本课程旨在开发基于开源框架的自动化软件分析技术，以了解和发现系统软件中常见但重要的软件缺陷和缺陷。通过本课程，学生将有机会应用和实践他们的系统编程技能和软件开发经验。特别是，学生们将设计和开发自动代码分析工具，以理解、发现和检测编程错误，从而提高软件质量。评估学生的依据是他们的技术能力、通过开源软件开发对软件分析的理解，以及他们在口语和写作方面的沟通技能、团队合作技能。

English Description

This subject is a practical subject designed for computer science and technology major. Software systems, which are deeply rooted in a wide range of industries and businesses, are so pervasive that we are often unaware of its presence until software bugs occur. A single bug can cause critical software failures, resulting in huge social and economic impacts. Despite the increasing attention and efforts in improving software reliability and security, modern complex systems (e.g. containing millions of lines of code) are still plagued with bugs. This subject aims to develop automated software analysis techniques based on an open-source framework to understand and discover common yet important software bugs and defects in system software. Through this subject, students will be given opportunity to apply and practice their system programming skills and software development experience. Particularly, the students will design and develop automated code analysis tools to understand, discover and detect programming errors to improve software quality. Students are assessed on the basis of their technical capabilities, understanding of software analysis via open-source software development, and their communication skills in speaking and writing, their team-working skills.

三、课程目标

1、知识目标：

1-1：掌握 C++程序的相关知识，重点理解抽象，封装，继承；熟练使用容器集合；理解指针和引用；熟练使用虚函数和多态。

1-2：理解 LLVM 编译器的相关知识和 IR，理解 IR 的 Scopes。掌握将 C 语言编译成 LLVM IR 的过程，理解 IR 的含义。

1-3：理解开源框架 SVF，理解使用 SVF 接口生成的 LLVM IR 中间图表示，理解用图表示程序的控制流和数据流。

1-4：使用 SVF 框架进行代码的污点检测分析。

2、能力目标：

2-1：运用系统编程技能解决复杂的软件工程问题。

2-2：应用系统的软件工程综合和设计过程来开发开源工具，通过开发开源软件工具学习编写高质量的源代码。

2-3：应用系统的方法管理软件工程项目，在团队中高效工作。

3、德育/素养目标：

3-1：软件程序设计与开发的过程中，应当遵守工程伦理的原则。

3-2：认识到创新能力的重要性。

3-3：能够熟练使用英语，使用口头或书面方式进行有效沟通和交流。

表 1 课程目标与毕业要求的对应关系

毕业要求	指标点	课程目标	关联 (H/M/L)
2、问题分析：能够应用数学、自然科学和工程科学的基本原理、方法和手段，识别、表达、并通过文献研究分析复杂工程问题，以获得有效结论。	指标点 2-1：专业知识掌握扎实，了解中国和世界经济领域及其研究的最新动态和发展趋势；能够应用数学、自然科学和工程科学的基本原理、方法和手段，分析、识别、表达本专业相关的复杂工程问题；	1-1、1-2、1-3、1-4	H
	指标点 2-2：能够应用数学、自然科学和工程科学的基本原理、方法和手段，针对实际复杂工程问题设计针对性的技术方案，并综合运用文献、科学理论和技术手段予以解决。	2-1、2-2、2-3	M
5、使用现代工具：能够针对复杂工程问题，开发、选择与使用恰当的技术、资源、现代工程工具和信息技术工具，包括对复杂工程问题的预测与模拟，并能够理解其局限性。	指标点 5-1：能够对本专业相关复杂工程问题进行建模与分析，理解获取相关信息参数的必要性与基本方法，并理解其局限性；	1-3 、 1-4 、 2-3	M
	指标点 5-3：能够针对本专业相关复杂工程问题，选择与使用恰当的技术、资源、现代工程工具和信息技术工具。	2-1、2-2、2-3、3-2、3-3	H
10、沟通：能够就本专业复杂工程问题与业界同行及社会公众进行有效沟通和交流，包括撰写报告和设计文稿、陈述发言、清晰表达或回应指令。具备一定的国际视野，能	指标点 10-1：能够就计算机领域相关复杂工程问题与业界同行及社会公众进行有效沟通和交流，能够通过口头或书面方式实现有效表达；	2-3 、 3-1 、 3-2、3-3	L

够在跨文化背景下进行沟通和交流。			

四、主要教学内容

（一）理论教学

第 1 章：Subject Overview and Introduction to Software Analysis （支撑教学目标 1-1、1-2、1-3、1-4、3-1、3-2、3-3）

【1】知识点：

(1) Goal of the Software Studio Subject.

(2) Introduction to Software Analysis.

(3) The Project of This Subject.

(4) Configure Programming Environment.

【2】学习目标：

理解：What is Software/Program Analysis.

掌握：Configure Programming Environment.

【3】重点内容：

(1) What is Software/Program Analysis.

(2) Configure Programming Environment.

【4】难点内容：

(1) What is Software/Program Analysis.

(2) The Project of This Subject.

第 2 章：C++ Programming Practices （支撑教学目标 1-1、2-1、2-2、2-3、3-1、3-2、3-3）

【1】知识点：

(1) The Characters of Object-Oriented Languages. C++ Classes and Objects.

(2) Containers/Collections.

(3) Pointers for Primitive Types. References for Primitive Types. Parameter Passing Using Pointers and References.

(4) C++ Inheritance. C++ Function Overriding. C++ Virtual Function and Polymorphism.

(5) C++ Programming Practice and Graph Algorithm.

【2】学习目标：

理解：Object-Oriented Programming, Pointers, Containers, and Classes.

掌握：Get Yourself Familiar with Integrated Development Environment (IDE). To practice coding and debugging skills and make necessary preparations for developing a software analysis tool in later assessments.

【3】重点内容：

(1) The Frame of a C++ Program.

(2) The Programming Style and Documentation.

(3) Coding and Debugging Skills.

【4】难点内容：

(1) Parameter Passing Using Pointers and References.

(2) C++ Virtual Function and Polymorphism.

第 3 章：LLVM Compiler and Graph Representation of Code （支撑教学目标 1-2、2-1、2-2、2-3、3-1、3-2、3-3）

【1】知识点：

(1) LLVM Compiler.

(2) LLVM Intermediate Representation (IR).

(3) Graph Representation of Code.

【2】学习目标：

理解：Understand open-source development methodology and to apply compiler fundamentals for source code analysis.

掌握 This task requires the student to start the first iteration of open-source development. The students will learn and apply their system programming skills in open-source tool implementation.

【3】重点内容：

To apply compiler fundamentals for source code analysis.

【4】难点内容：

(1) Compiler Fundamentals.

(2) Graph Representation of Code.

第 4 章：Control-Dependence and Data-Dependence （支撑教学目标 1-3、2-1、2-2、2-3、3-1、3-2、3-3）

【1】知识点：

(1) Control-Dependence: The execution order between two program statements/instructions.

(2) Data-Dependence: The definition-use relationships between variables.

【2】学习目标：

理解：Understand Control-Dependence. Understand Data-Dependence.

掌握：Implement a Context-Insensitive ICFG Traversal. Implement Andersen's Pointer Analysis.

【3】重点内容：

Source Code Analysis.

【4】难点内容：

Validating the Quality and Correctness of the Developed Software Analysis Tool.

第 5 章：Taint Analysis （支撑教学目标 1-4、2-1、2-2、2-3、3-1、3-2、3-3）

【1】知识点：

(1) Use what we have learned about control- and data-dependence to develop an information flow checker to validate tainted flows from a source to a sink.

(2) Configuring Sources and Sinks for Taint Analysis.

【2】学习目标：

理解：Detect Information Leakage. Detect Code Vulnerability.

掌握：Aims to reason about the control and data dependence from a source (statement/node) to a sink (statement/node).

【3】重点内容：

Static Taint Analysis.

【4】难点内容：

Information Flow Tracking Analysis.

【5】思政案例：

The goal of ideological and political education is to promote the spirit of scientists, inspire and lead students to tirelessly strive for a new journey. It also aims to encourage students to learn about and promote the spirit of scientists during the teaching process. A person without spirit cannot stand firm, and a country without spirit is not strong. In the wave of the new era, guided by the spirit of scientists, we shoulder historical responsibilities, explore and innovate, forge ahead, and make greater contributions to the great rejuvenation of the Chinese nation.

五、教学安排与学时分配

表 3 学时分配表

讲　　课　　内　　容		学时				
		讲课	实验	上机	设计	课外
第 1 章	Subject Overview and Introduction to Software Analysis	4	0	0	0	0
第 2 章	C++ Programming Practices	8	0	0	0	0
第 3 章	LLVM Compiler and Graph Representation of Code	4	0	0	0	0
第 4 章	Control-Dependence and Data-Dependence	20	0	0	0	0
第 5 章	Taint Analysis	12	0	0	0	0
小计		48	0	0	0	0
总计	48					

六、教学方法

1、Multimedia-based lecturing.
2、Combining theory with industrial practical problems.
3、Knowledge delivery with ethic education.
4、Project-driven with guidance, materials, examples, and project specifications.

七、成绩评定

1、The assessment process consists of behavior grades, mid-term grades, final grades and bonus grades. The result type is marks, and the full marks is 100 points. Behavior grades account for 30%, mid-term grades account for 32%, and final grades account for 38%. The bonus task is optional and contributes 5 points (The total marks of this subject will be capped at 100).
2、 The behavior part documents students' class performance and attendance. No score will be awarded for absences.
3、The mid-term part will be evaluated by students' quizzes on CC Canvas.
4、The final-term part will be evaluated based on students' coding assignments on CC Canvas..

八、评价及持续改进

1、课程负责人根据课程教学内容与人才培养目标组织课程团队讨论并修改教学大纲，报分管教学工作副院长审核后由执行院长批准。
The subject coordinator shall be responsible for the syllabus discussion and improvement, and the revised version shall be submitted to deputy dean (teaching affairs) for reviewing then to executive dean for approvement.
2、课程负责人根据课程教学内容、课堂教学效果以及成绩分布，对课程教学方法和成绩评定环节进行改进，并同步优化评定办法。
The subject coordinator shall revise the syllabus based on the teaching content, effect and result distribution while optimize the assessment measures.

撰写：　　程绍辉、张旭

审核：　　韩鹏

批准：　　史闻博

日期：　　2023.07.05