

第二批国家级一流本科课程申报书

(虚拟仿真实验教学课程)

课程名称：C++函数参数传递可视化虚拟仿真实验

专业类代码：0809

负责人：何婷婷

联系电话：13971447897

申报学校：华中师范大学

填表日期：2021 年 04 月 23 日

推荐单位：华中师范大学

中华人民共和国教育部制

二〇二一年四月

填报说明

1.专业类代码指《普通高等学校本科专业目录（2020）》中的专业类代码（四位数字）。

2.文中○为单选；□可多选。

3.团队主要成员一般为近5年内讲授该课程教师。

4.文本中的中外文名词第一次出现时，要写清全称和缩写，再次出现时可以使用缩写。

5.具有防伪标识的申报书及申报材料由推荐单位打印留存备查，国家级评审以网络提交的电子版为准。

6.涉密课程或不能公开个人信息的涉密人员不得参与申报。

1. 基本情况

实验名称	C++函数参数传递可视化虚拟仿真实验	是否曾被推荐	○是√否
实验所属课程 (可填多个)	面向对象程序设计 (C++)、高级语言程序设计 (C)		
性质	○独立实验课 √课程实验		
实验对应专业	计算机科学与技术		
实验类型	√基础练习型 ○综合设计型 ○研究探索型 ○其他		
虚拟仿真必要性	<input type="checkbox"/> 高危或极端环境 <input type="checkbox"/> 高成本、高消耗 <input type="checkbox"/> 不可逆操作 √大型综合训练		
实验语言	√中文 ○中文+外文字幕 (语种) ○外文 (语种)		
实验已开设期次	共 3 次： 1. 时间：2020 年 3 月 - 2020 年 6 月、人数：45 人 2. 时间：2020 年 9 月 - 2021 年 1 月、人数：120 人 3. 时间：2021 年 3 月 - 2021 年 4 月、人数：380 人		
有效链接网址	http://csvlab.ccnu.edu.cn		

2. 教学服务团队情况

2-1 团队主要成员 (含负责人, 总人数限 5 人以内)								
序号	姓名	出生年月	单位	职务	职称	手机号码	电子邮箱	承担任务
1	何婷婷	1964-04	华中师范大学	院长	教授	13971447897	tthe@ccnu.edu.cn	在线教学服务人员
2	张 勇	1978-09	华中师范大学		副教授	15802766977	ychang@ccnu.edu.cn	在线教学服务人员
3	刘 明	1967-12	华中师范大学	副院长	教授	13807127439	lium@ccnu.edu.cn	在线教学服务人员
4	王敬华	1965-10	华中师范大学	副院长	副教授	13707103274	jhuawang@126.com	在线教学服务人员
5	沈显君	1973-12	华中师范大学	系主任	教授	13667126948	xjshen@ccnu.edu.cn	在线教学服务人员

2-2 团队其他成员						
序号	姓名	出生年月	单位	职务	职称	承担任务
1	张茂元	1975-06	华中师范大学	副院长	教授	在线教学服务人员
2	杨进才	1967-11	华中师范大学		教授	在线教学服务人员
3	魏开平	1962-10	华中师范大学		副教授	在线教学服务人员
4	赵尔敦	1972-08	华中师范大学		副教授	在线教学服务人员
5	胡珀	1980-05	华中师范大学		副教授	在线教学服务人员
6	涂新辉	1979-12	华中师范大学		副教授	在线教学服务人员
7	罗昌银	1983-02	华中师范大学		讲师	在线教学服务人员
8	彭熙	1978-11	华中师范大学		讲师	在线教学服务人员
9	陈曙	1981-01	华中师范大学		讲师	在线教学服务人员
10	高丽	1964-08	华中师范大学		正高	技术支持人员
11	黄枫	1990-07	华中师范大学		中级	技术支持人员
12	吴德钰	1990-09	华中师范大学		中级	技术支持人员
13	王婷婷	1991-12	华中师范大学		中级	技术支持人员
14	孙树园	1991-09	华中师范大学		中级	技术支持人员
15	周伟	1980-03	华中师范大学		中级	技术支持人员
团队总人数：20 人 其中高校人员数量：20 人 企业人员数量：0 人						

2-3 团队主要成员教学情况（限 500 字以内）

（近 5 年来承担该实验教学任务情况，以及负责人开展教学研究、学术研究、获得教学奖励的情况）

负责人何婷婷长期从事计算机教学与研究，为华师计算机科学与技术国家级一流本科专业负责人、华中师范大学桂苑名师、计算机软件课程校级优秀教学团队负责人，湖北省高教学会计算机教育专委会主任。主持教育部、湖北省教学研究项目多项，获得湖北省教学成果一等奖、二等奖、三等奖各一项。主持国家自然科学基金重点项目、国家社科基金重大计划项目各一项，获得湖北省科技进步一等奖、二等奖、文化部创新奖各一项，获得全国优秀科技工作者、湖北省有突出贡献中青年专家等荣誉。

近五年来，团队主要成员承担了 C 语言、C++、Java、C#、Python 等主要程序设计语言的课堂教学和实验教学，建设有国家级实验教学示范中心、湖北省实验教学中心等，主持了湖北省省级教学研究项目 9 项，建设了湖北省精品课程 3 门，获得湖北省高等学校教学成果奖 1 项，编撰普通高等教育“十二五”国家级规划教材 2 部，指导大学生科研创新并获得第十五届“挑战杯”国家级一等奖、中国大学生计算机设计大赛国家级一等奖、中国大学生程序设计大赛国家级一等奖、中国机器人大赛国家级一等奖、全国高校互联网应用创新大赛国家级一等奖、蓝桥杯全国软件和信息技术专业人才大赛国家级一等奖等共计 42 项，在实验教学方面取得了丰硕的成果。

注：必要的技术支持人员可作为团队主要成员；“承担任务”中除填写任务分工内容外，请说明属于在线教学服务人员还是技术支持人员。

3. 实验描述

3-1 实验简介（实验的必要性及实用性，教学设计的合理性，实验系统的先进性）

信息化技术的飞速发展给计算机软件人才的培养带来了极大的挑战，作为软件人才培养的基石，计算机程序设计语言的教学至关重要。**C++**程序设计语言是目前最流行的面向对象程序设计语言之一，以**C**语言为基础发展而来，在计算机软件开发领域具有举足轻重的作用，几乎是各大高校计算机专业的必修课程。但是，由于计算机系统构建了一个完全不同于现实世界的复杂的虚拟环境，其内部运行过程构建于高度复杂和层次化的软件和硬件系统之上，具有不可见性，难以被直观地观察到，从而使得程序语言的学习者难以理解程序代码运行的内部过程和原理，只能从计算机运行的视角来想象和推测，这给**C++**程序设计语言的实验教学带了极大的挑战，特别是**C++**程序设计语言主要面向计算机基础和计算机思维尚不成熟的低年级学生。

程序执行过程不可见的问题给**C++**教学带来的负面作用主要体现在以下几个方面。

（1）程序的执行过程成为了一个黑盒，其执行逻辑和过程不可见，学习编程十分困难；（2）低年级学生缺乏对计算机体系结构、内存分配等基础知识的融会贯通，教学效果十分有限；（3）**C++**函数调用和参数传递涉及内存中重要的数据结构“栈”，但不可见性让学生难以理解其中的机制和原理；（4）**C++**中的指针、引用、动态内存分配、取地址操作等知识与内存密切相关，程序执行的不可见性使得相关知识点的教学晦涩难懂。因此，如何让计算机的“虚拟世界”变成直观可见，对**C++**语言的教学将具有十分重要的意义。

目前已存在一些计算机可视化方面的技术和软件，例如针对少儿编程教学的**Scratch**。该软件是由美国麻省理工学院(MIT)设计开发，通过可视化的拖拽积木完成编程。但其可视化仅仅针对编程这个环节，并没有涉及计算机内部执行过程的可视化，且其编程主题和范围都较为简单，主要针对一些固定简单的少儿游戏。

为此，教学组采用虚拟仿真技术，研究开发了**C++**语言可视化执行虚拟仿真实验平台。该平台能够动态和可视化的展示**C++**程序指令运行时程序变量在计算机内存中的分配、地址以及值的变化，能够让学生更加直观地观察到每条**C++**程序指令运行时对应的内存状态，帮助学生更加深刻地理解**C++**程序和计算机内存的运行原理。同时，该平台还提供了**C++**程序编写、修改、编译、调试和程序片段切换等交互式实验功能，为学生提供了一个功能完善的**C++**在线可视化实验平台（该平台同时也支持**C**语言的在线实验）。同时，教学组以此平台为基础，以“**C++**函数参数传递实验”为例，针对性地设计开发了“**C++**函数参数传递可视化虚拟仿真实验”，解决了程序运行时的不可见性问题，使教学具有直观性、互动性和探索创新性，符合学生的认知规律，便于学生接受；基于**Web**的可视化实验平台设计合理，容易实施和部署，兼容性好，随时随地的在线实验教学突出了学生的教学主体地位，突破了时空限制，学生只需要一台带浏览器的便携设备即可随时开展实验。

该实验系统的先进性主要体现在三个方面。（1）该实验的可视化设计完全由**C++**

程序驱动，而不是固定的脚本驱动。传统的虚拟仿真实验往往针对特定实验内容设计固定的脚本，然后通过 3D、VR 等技术来实现其过程的可视化。这种传统制作方案灵活性较差，对不同实验内容需要单独设计和制作，甚至实验脚本的升级改动都会导致整个实验系统重新制作，因此其成本居高不下。而该实验的可视化过程并不是由固定的脚本驱动，而是直接由 C++ 程序驱动，程序改动后其执行可视化过程会自动根据程序逻辑而调整，因而其将来升级和扩展的成本较低，可扩展性较好。

(2) 教学组研发的 C++ 语言可视化执行虚拟仿真实验平台虚实结合，具有良好的通用性和可扩展性。该平台能够可视化执行任何 C++ 程序，并将真实的程序运行信息和动态可视化的虚拟仿真相结合，由服务器编译、调试 C++ 程序来获取程序执行时的真实信息并传递给客户端，由客户端的可视化技术来虚拟仿真计算机内部状态。因此，该平台不仅可以用于开发其他的 C++ 实验项目，还能应用于 C++ 课堂教学、学生课后练习和探索等其他教学场合。

(3) 该实验对提升学生整体计算机系统能力具有重要意义。该实验系统不仅展示了 C++ 函数参数传递机制，还包含了计算机内存、程序运行时的内存分布、数据结构中的栈、程序在整个计算机内的执行过程等相关知识，将 C++ 语言、操作系统、数据结构、计算机体系结构等多门课程的知识融会贯通，促进了学生对计算机整体运行机制的理解和认识，有助于提升学生的整体计算机系统能力。

3-2 实验教学目标（实验后应该达到的知识、能力水平）

(1) 掌握 C++ 函数之间参数按值传递、按地址传递、按引用传递的差异；

在使用 C++ 语言进行程序编写时，函数的调用是非常普遍的。函数的调用又可分为无参调用和有参调用，通常我们需要将一个或多个参数传递至函数内部进行相关的计算操作，这种调用方式就是有参调用。在 C++ 语言中，有参调用的参数传递方式有值传递、引用传递以及指针（地址）传递三种，这三种传递方式有着很大的区别，且不同的传递方式对结果会产生不同的影响，深入理解和掌握这三种参数传递规则是学习 C++ 编程必不可少的步骤。

具体而言，在使用值传递时，形参和实参是两个独立的变量，各自在内存中拥有不同的内存空间，系统将实参的值进行拷贝然后赋值给形参，在被调函数中修改形参的值不会影响实参本身；使用引用传递时，形参是系统给实参起的一个别名，形参代表的就是实参本身，因而系统不会给形参再额外分配内存，并且在被调函数中修改形参的值将会改变实参本身；使用指针（地址）传递时，形参必须是指针变量类型，形参中存放的是实参的地址，通常也称该指针（地址）指向了实参，通过 * 运算符可以对指针中的内容进行操作，并且该操作将改变实参的值。

完成实验后，学生应能够快速区分这三种参数传递方式，了解它们之间的差异及其背后的原理，并在实际代码编写过程中熟练使用每种参数传递方法。

(2) 掌握 C++ 函数之间参数传递的机制和顺序；

在函数调用时，系统首先会将调函数加载到栈顶，然后从左到右依次检查调函数的每一个参数项，并且依次对调函数的形参进行初始化赋值。初始化赋值时，根据形参的数据类型不同，赋值方式也不同。对于值传递，系统会将实参的值进行拷贝，然后赋值给形参；对于引用传递，形参即代表实参本身，不再额外分配内存；对于地址传递，形参会使用实参的内存地址进行初始化赋值。

完成实验后，学生应了解被调函数函数加载的位置，掌握初始化形参变量时参数赋值的过程及其顺序，能够设计代码来验证形参初始化规则。

(3) 掌握 C++ 函数中形参、局部变量在栈中的内存分配机制；

一个函数被调用时，系统会为其形参和局部变量分配内存，且这部分的内存分配将会发生在栈区。在栈区分配的内存都有其对应的作用域及寿命，作用域是指在哪一区域中该变量是有效的，寿命是指在什么时候该变量所分配的内存被系统回收掉。当一个变量不在其作用域中，那么对这个变量的任何操作都是非法的，一旦在超出其作用域的位置使用该变量，程序将会报错。此外，在一个变量的寿命结束之后，再次使用该变量也会使程序出错。因而，了解形参和局部变量在内存中的分配机制，有助于学生正确使用这些变量，避免编写的程序在该方面报错。

完成实验后，学生应熟悉函数中的形参和局部变量在内存中存放的方式，掌握形参和局部变量的作用域及其寿命。

(4) 掌握 C++ 程序运行时的内存分配机制和原理；

C++ 程序在运行时内存分配方式可以分为静态内存分配和动态内存分配两种。这两种分配方式分配出的内存是有差别的，前者分配出的内存存在栈区，而后者分配出的内存存在堆区。在栈区分配的内存一般是临时的（main 函数中分配的内存会在程序结束时回收），即其中某个函数或者变量寿命结束，系统会将其从栈中弹出并回收其占用的内存空间，这个过程不需要程序员自己干预。在堆中分配的内存会伴随程序直至结束，如果想要提前释放掉某块内存空间，则需要程序员编写一段释放该内存的代码，并且在释放结束之后，之前占用该内存的变量也不得再使用。特别是释放掉指针变量所指向的内存后，再次操作该指针变量是相当危险的！因为该内存释放后系统会将其分配给其他程序使用，那么再次对该指针进行操作可能会导致系统崩溃。

另外，静态内存分配和动态内存分配在代码层面的实现是不同的。最常规的直接定义某个变量使用的就是静态内存分配方式，该变量所占用的内存大小在程序运行过程中不会再发生改变（内存中的值可变），且不同的数据类型对应了不同的内存大小，这将由编译器自动计算完成。动态内存分配在 C++ 中的实现需要使用关键词 new（或者使用 C 语言的 malloc 函数），以这种方式分配出的内存存在程序运行过程中是可变的，且这部分内存需要由程序员手动释放，如果没有及时

释放，可能会导致系统内存耗尽。

完成实验后，学生应了解两种内存分配方式的特点，掌握每种分配方式其背后的原理，并且能够从原理出发阐述这两种内存分配方式的区别，最后还要能够在实际的代码编写过程中利用这两种分配方式来定义不同的变量。

(5) 提高学生主动探索学习的能力；

C++语言可视化执行虚拟仿真实验平台能够将任何 C++程序的执行过程可视化，从计算机内部揭示程序运行的原理和过程，能够更加直观地显示程序执行的逻辑，帮助学生发现程序中的错误。当学生编写的程序出现问题时，可以借助本平台来可视化执行以探索其中的原因，帮助学生理解 C++语言中的知识点，提高学生主动探索学习的能力。

3-3 实验课时

(1) 实验所属课程课时：面向对象程序设计 63，高级语言程序设计 48

(2) 该实验所占课时： 2 课时

3-4 实验原理

(1) 实验原理(限 1000 字以内)

计算机系统的高度复杂性导致其内部原理很难被直接观察和理解，具有不可见性。一方面，学生在学习 C++程序设计语言时难以观察到高级指令运行后的效果，只能依靠想象和推测来进行理解和掌握，学习难度比较大；另一方面，教师在讲授的时候也难以用语言描述清楚，通常要通过图表来展示计算机内存的内部状态，但是图表是静态的，难以展示复杂的动态程序运行的效果。同时，程序逻辑通常较为复杂，程序上的些许变化都会导致运行上的巨大差异，从而导致实验教学效果十分有限。为此，教学组利用 C++程序调试工具 Valgrind、Linux 环境下的应用容器引擎 Docker、浏览器端的可视化技术 HTML5 和 SVG 等技术，设计开发了 C++语言可视化执行虚拟仿真实验平台。同时，教学组以 C++课程实验中最典型的“函数参数传递”为例，基于该平台设计开发了“C++函数参数传递可视化虚拟仿真实验”。

该实验平台主要由两个部分组成：真实的 C++程序运行环境和虚拟的计算机内部状态可视化仿真，具有虚实结合、能实不虚的特点。

真实的 C++程序运行环境是在服务器上基于类 Linux 系统下的应用容器引擎 Docker 创建的一个安全的“沙箱”环境。在“沙箱”中，C++源代码通过 GCC 编译，然后由 Valgrind 调试工具来运行并收集计算机内部状态信息，最后将运行时的计算机内部状态信息以 JSON 格式返回给浏览器端。

虚拟的计算机内部状态可视化仿真则是在浏览器端，运用 HTML5、SVG、JavaScript 等技术，调用服务端的 C++程序运行环境以获得 JSON 格式的计算机

内部状态信息，并将每一行程序执行时的内部状态绘制在网页上，通过高度的抽象化来展示计算机内存中“堆、栈、全局数据区”的结构信息。

“C++函数参数传递可视化虚拟仿真实验”以该实验平台为基础，共设计了两组实验（基础性实验和扩展性实验）和一组在线知识点巩固测试。

基础性实验由三个程序片段组成，分别为按值传递的函数调用、按指针传递的函数调用和按引用传递的函数调用。扩展性实验则属于综合性实验，将按值传递、按指针传递、按引用传递三者混合使用，以考察学生对函数参数传递本质的理解，具有较高的难度和深度。同时，扩展实验还允许学生在线修改程序源代码并可视化执行，从更广泛的角度来探索函数参数传递的本质和内涵。

（2）**核心要素仿真设计**（对系统或对象的仿真模型体现的客观结构、功能及其运动规律的实验场景进行如实描述，限 500 字以内）

本实验的核心要素仿真设计主要由前端和后端两个部分组成，如图 1 所示。

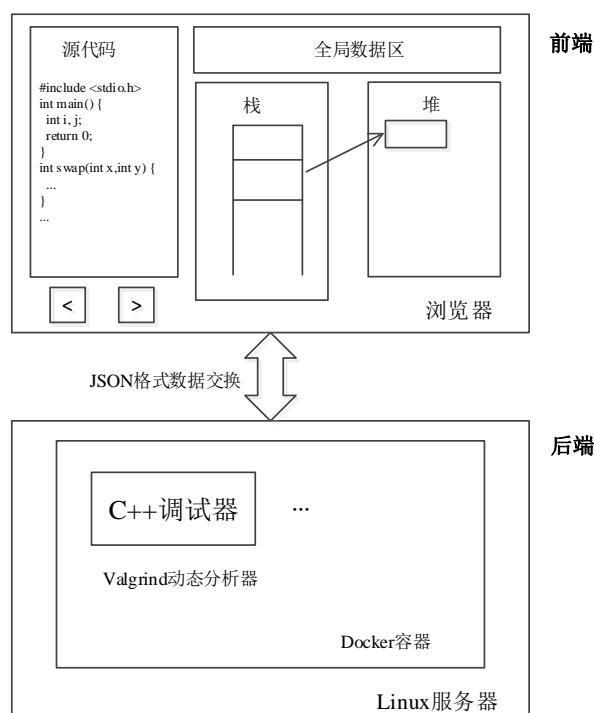


图 1 核心要素仿真设计：前端和后端

后端是构建在类 Linux 服务器上的一个真实的 C++ 程序运行环境，其功能主要是接收学生的 C++ 源代码，然后构造一个安全运行环境来编译和调试 C++ 程序代码，并将程序调试的信息以 JSON 格式返回给请求者。

前端则是构建在浏览器端的一个可视化虚拟仿真环境，其作用主要是根据程序调试的 JSON 数据，可视化的绘制每一条程序运行后计算机内存中堆、栈、全局数据区中变量、值及其关系，以高度抽象的形式可视化的展示计算机内部状态，将不可见的计算机内部信息虚拟仿真为可见的状态，并允许学生通过程序与

其进行交互操作。

该实验平台的设计既包含有“实”的部分，也包含有“虚”的部分，具有虚实结合的优点。“实”的部分体现在平台的后端，即 C++ 程序会被真实地编译、运行，并获取真实的计算机内部状态信息，如变量的内存地址、值等。“虚”的部分体现在平台的前端，即在浏览器端，“不可见”的计算机内部状态会被可视化地虚拟仿真出来，并能随着程序的执行动态演变。由于计算机的高度复杂性，过多的细节展示会干扰学生对逻辑的理解，因此，在可视化的时候，平台针对 C++ 课程的教学需求，对计算机内部状态和逻辑结构进行了高度的抽象化，虚拟仿真的是高度抽象化的逻辑结构，可视化展示的内容是真实的计算机内部数据信息，从而达到了虚实结合的特点。

3-5 实验教学过程与实验方法

本实验以 C++ 函数中三种不同类型的参数传递为主线，采用虚拟仿真技术对函数调用过程中，实参和形参的内存分配、回收、数据交换以及栈的运作原理进行全方位地展示，将不可见的计算机内部逻辑运作机制以可视化的方式呈现，以期让学生能够快速、直观地掌握栈在 C++ 函数调用中的作用，掌握计算机内存的逻辑结构和分配机制等知识点内容。

(1) 实验教学过程

1. 问题式教学法的相关问题知识准备。让学生查阅资料，了解 C++ 程序中基本数据类型、指针变量和引用变量的基本概念和使用方法；了解 C++ 程序中函数定义、形参、实参等基本概念、区别和使用方法；了解计算机内存的分配策略和形式，特别是内存中栈的基本概念及其在函数调用中的作用；了解函数调用时实参与形参的相互作用机制，特别是栈在参数传递中的作用；了解 C++ 函数中局部指针变量在栈中的分配策略等，为正确回答问题做好知识储备。

2. 虚拟仿真实验训练。以科学问题为导向，基于前期的知识准备，让学生操作虚拟仿真软件，从不同的角度掌握函数参数的存储、实参与形参的交互过程、不同类型的形参的特点、运算及其结果，掌握函数参数传递的基本规律和内置原理；掌握计算机内存中栈、堆和全局数据区的分布、分配策略、特点及其在 C++ 程序执行时的作用和用法等，旨在深入理解相关科学问题，掌握计算机操作系统、高级语言程序设计和计算机组成原理等相关知识，拓展知识面。

3. 针对科学问题的讨论交流：在知识储备、问题式教学、实验操作基础上，针对 C++ 函数参数传递的机制和不同类型参数在传递时的特点凝练讨论专题，让学生分组讨论，写出讨论报告，拓展理论知识，引发思考和系统总结。

4. 探究总结并提出创新性问题：在问题式教学、虚拟实验训练、专题讨论基础上，针对相应的科学问题进行总结和研究，分小组以科研小论文的形式撰写研究报告。并提出新的科学问题，锻炼文献综述、学术观点总结及科学问题凝练

等能力，最终培养学生的创新能力。

（2）实验方法

本实验采用问题式教学法、虚拟仿真实验教学法、讨论式教学法、探究式教学法等教学方法，也代表教学过程层层深入的4个环节。

1. 问题式教学法

问题式教学方法是设置与教学内容相关的科学问题、让学生带着问题查阅文献，了解科学问题的相关知识，为虚拟实验操作打基础。本实验设置的科学问题有：①计算机内存中栈的分配和回收机制；②计算机内存中全局变量区的分配机制；③计算机内存中堆的分配和回收机制；④ C++程序中函数调用与栈的切换；⑤ C++函数中形参的含义及其被调用时在栈中的分配时间和机制；⑥ C++函数被调用时形参在栈中的分配机制；⑦ C++函数中局部变量在栈中的分配时间和机制；⑧ C++中为指针申请额外内存空间的分配机制；⑨ C++中引用类型的本质及其含义；⑩ C++函数调用时实参和普通形参、指针型形参、引用型形参的传递机制；

2. 虚拟仿真实验教学法

虚拟仿真实验教学法是将原来不能做、做不好的实验通过虚拟仿真实验方式来完成，加强学生相关实验能力的训练。该实验通过虚实结合的技术手段将原来不可见的计算机内部状态以可视化、动态、直观的形式呈现出来，让学生在可视化的环境中进行实验学习，简单直观的观察整个计算机内存的分布特征、栈的函数调用中的作用和机制、每行程序代码对内存的影响、不同数据类型在内存中的分配机制等信息，体验和验证问题式教学环节中各科学问题涉及相关知识，加深对上述问题和相关知识点的理解掌握。

3. 讨论式教学法

该实验以虚拟仿真实验平台为依托，针对实验教学中存在的问题师生开展讨论交流，通过班级讨论和小组讨论相结合的方式进行，并给出各自的书面答案，以收到答疑解惑的效果。本实验设置的讨论问题如：① 形参与局部变量有什么区别和联系；② 指针和引用类型的实参赋值给形参后，形参能够执行哪些操作，这些操作是如何影响实参和实参所指向的存储空间；③ 形参进行数据交换时，被交换的形参或局部本身以及所指向空间的区别和联系；④ 指针变量的操作符*和普通类型的操作符&的使用在逻辑上差异和联系；⑤ 各种类型的实参变量与形参进行参数传递的机制。通过专题讨论让学生学会总结提高，综合掌握C++函数传递的相关理论和实验知识，并通过知识拓展，了解计算机内存的逻辑结构，特别是栈的结构、分配和回收机制以及在函数调用中的作用。

4. 探究式教学法

探究式教学法是在问题式教学法、虚拟仿真实验体验、讨论式教学法等基础上进行，让学生针对相关知识点进行系统的总结和拓展，在理论和实践相结合的高度对相关问题进行深入研究，形成一定水平的理论总结和继续深入研究的科学问题。旨在提高学生的文献综述、实验总结、知识拓展及科学问题凝练等科研思维和创新能力。本实验探究分析的问题包括① 函数参数传递时，实参与形参类型不完全一致时的处理机制；② 函数参数传递与缓冲区溢出漏洞的关系探讨；③ C++程序中的代码段、数据段、栈、堆等逻辑空间在内存中的分布及其特点；④ 栈空间的作用机制以及在编程时需要注意的问题总结；⑤ 从可视化的 C++程序运行来探讨内存泄露问题，以及如何避免内存泄露。

3-6 步骤要求（不少于 10 步的学生交互性操作步骤。操作步骤应反映实质性实验交互，系统加载之类的步骤不计入在内）

（1）学生交互性操作步骤，共 13 步（主要步骤）

步骤序号	步骤目标要求	步骤合理用时	目标达成度赋分模型	步骤满分	成绩类型
1	值传递程序调试	2 分钟	启动调试：5 分 未启动调试：0 分	5	✓ 操作成绩 ✓ 实验报告 <input type="checkbox"/> 预习成绩 ✓ 教师评价报告
2	执行值传递函数	2 分钟	调用值传递函数：5 分 未启动调试：0 分	5	
3	回答值传递执行结果	5 分钟	回答正确：10 分 回答错误：0 分	10	
4	引用传递程序调试	2 分钟	启动调试：5 分 未启动调试：0 分	5	
5	执行引用传递函数	2 分钟	调用引用传递函数：5 分；未启动调试：0 分	5	
6	回答引用传递执行结果	5 分钟	回答正确：10 分 回答错误：0 分	10	
7	指针传递程序调试	2 分钟	启动调试：5 分 未启动调试：0 分	5	
8	执行指针传递函数	2 分钟	调用指针传递函数：5 分；未启动调试：0 分	5	
9	回答指针传递执行结果	5 分钟	回答正确：10 分 回答错误：0 分	10	
10	回答混合参数传递结果	5 分钟	回答正确：10 分 回答错误：0 分	10	
11	回答指针参数交换结果	5 分钟	回答正确：10 分 回答错误：0 分	10	
12	在线理论知识测试	20 分钟	随机抽取 10 题 每题 10 分，共计 20 分	20	
13	自定义扩展	20 分钟	开放实验，由学生自由	附加	

	实验		修改程序并进行可视化运行，由教师评分。	分： 15	
--	----	--	---------------------	----------	--

说明：扩展实验分为推荐扩展实验和自定义扩展实验，后者为探索性实验，属于附加分，由学生选做。以上步骤仅为关键性操作步骤。

(2) 交互性步骤详细说明

本实验主要可分为三个部分：基础实验、扩展实验和理论测试。其中，扩展实验属于开放式实验，学生运行推荐的扩展实验程序，也可以自由地在线修改程序并可视化执行。主要的交互性步骤详细说明如下。

基础实验部分

步骤一：学生访问该虚拟仿真实验网站（http://csvlab.ccnu.edu.cn），输入账号和密码进入虚拟仿真实验系统，在实验教学系统里能看到实验相关的简介、实验目标、开始实验和实验成绩。点击开始实验就能看到本系统的三个功能模块，点击基础实验。



图 2 实验登录和实验内容

步骤二：基础实验部分设置了三个演示实验，分别对应值传递实验、引用传递实验以及指针传递实验。每个实验中设计了相应知识点的代码实例，通过点击上方每个实验的按钮可以切换实验内容。首先点击值传递实验。



图 3 基础实验界面

步骤三：在值传递实验中，系统给出了一个 swap 函数，参数传递类型为值传递，通过点击下一步可以单步执行当前程序，左边代码区域的白色箭头指示上一步已执行的代码，红色箭头指示当前准备执行的代码。同时右边的数据区将显示每个变量在内存中存储的位置以及其对应的值。如图 4 所示，该系统可视化展示了程序执行到 main 函数调用 swap 函数时的内存状态，此时 swap 函数被加载到

栈中，准备为其形参进行初始化赋值。由于是参数传递方式是值传递，swap 函数的形参将会使用 main 函数中的实参的值拷贝进行赋值。

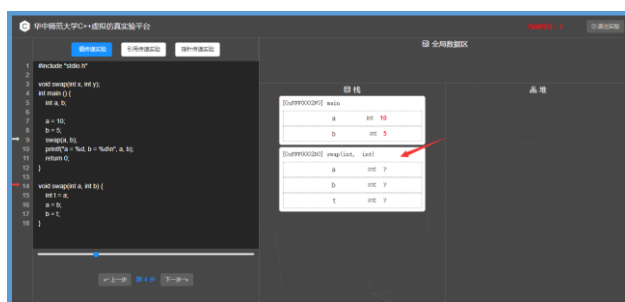


图 4 按值传递实验过程

步骤四：swap 函数执行完毕后，该函数将会从栈中弹出，swap 函数中的所有局部变量及其所占用的内存空间都会被系统回收。当程序执行到 printf 函数时，系统会弹出一个输入框检查学生对知识点的掌握情况，学生需要在输入框中分别填入程序执行到目前为止，main 函数中变量 a，b 所存放的值，根据学生填入的结果与正确结果进行核对，系统会自动完成对该知识点的考核和评分。

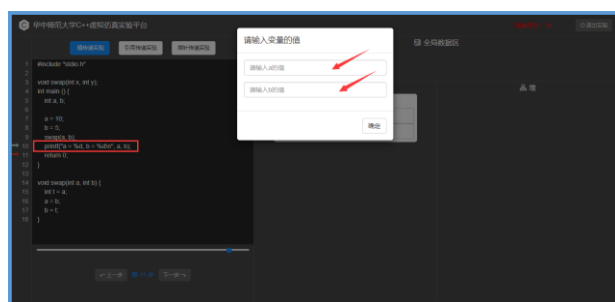


图 5 按值传递实验时的问题交互

步骤五：值传递实验结束之后，点击上方的引用传递实验按钮，进入引用传递实验。引用传递实验中，系统给出了一个 swapByRef 函数，该函数的参数传递类型为引用传递。同样地，学生通过点击下一步按钮单步执行该程序，并且变量值的变化情况将会在右边的数据区内显示出来。

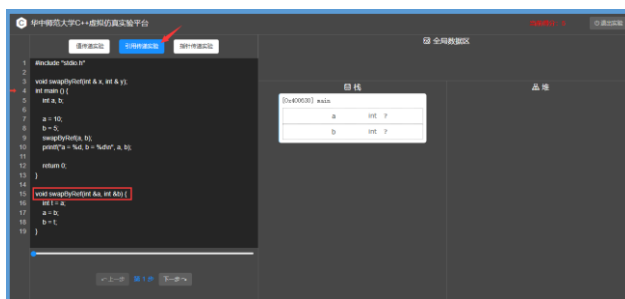


图 6 按引用传递实验界面

步骤六：当程序开始调用 swapByRef 函数时，白色箭头指向该函数的被调用处，红色箭头指向该函数的函数名。swapByRef 函数被加载到数据区的栈顶，准备为其形参进行初始化赋值，由于该函数的参数传递类型为引用传递，swapByRef 函

数的形参都将代表原实参本身。

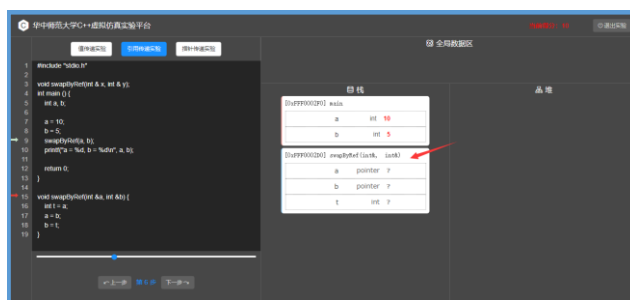


图 7 按引用传递实验过程

步骤七：swapByRef 函数执行完毕后，该函数将会从栈中弹出，swapByRef 函数中的所有局部变量及其所占用的内存空间都会被系统回收。当程序执行到 printf 函数时，系统会弹出一个输入框检查学生对知识点的掌握情况，学生需要在输入框中分别填入程序执行到目前为止，main 函数中变量 a，b 所存放的值，根据学生填入的结果与正确结果进行核对，系统会对本次实验进行打分。

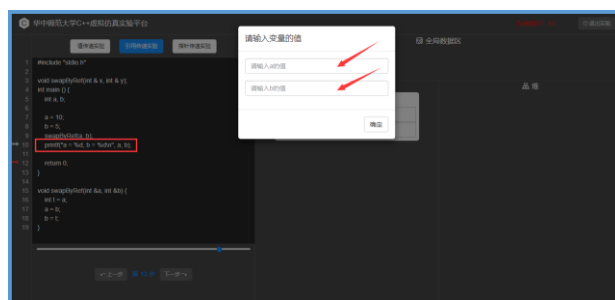


图 8 按引用传递实验时的问题交互

步骤八：引用传递实验结束之后，点击上方的指针传递实验按钮，进入指针传递实验。指针传递实验中，系统给出了一个 swapByPointer 函数，该函数的参数传递类型为指针传递。同样地，学生通过点击下一步按钮单步执行该程序，并且变量值的变化情况将会在右边的数据区内显示出来。

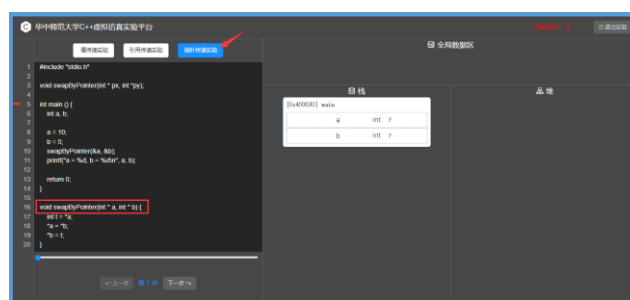


图 9 按指针传递实验界面

步骤九：当程序执行到 swapByPointer 函数内部时，右边的数据区展示了每个指针所指向的实参。由于指针 a 接收了实参 a 的地址，则指针 a 指向 main 函数中的实参 a；指针 b 接收了实参 b 的地址，则指针 b 指向 main 函数中的实参 b。白色箭头已完成的代码表示将 b 指针所指的实参的值赋值给 a 指针所指向的实

参，故在 main 函数中可以看到实参 a 的值已经从 10 变成了 5。



图 10 按指针传递实验过程

步骤十：swapByPointer 函数执行完毕后，该函数将会从栈中弹出，swapByPointer 函数中的所有局部变量及其所占用的内存空间都会被系统回收。当程序执行到 printf 函数时，系统会弹出一个输入框检查学生对知识点的掌握情况，学生需要在输入框中分别填入程序执行到目前为止，main 函数中变量 a，b 所存放的值，根据学生填入的结果与正确结果进行核对，系统会对本次实验进行打分。

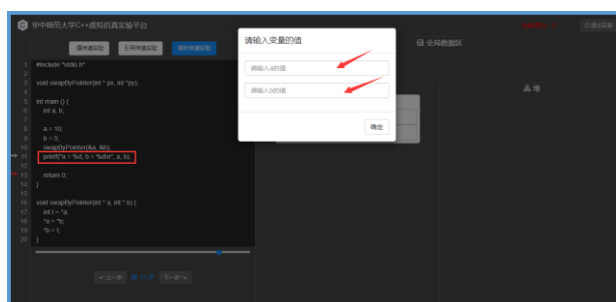


图 11 按指针传递实验的问题交互

扩展实验部分

步骤一：切换到扩展实验部分，扩展实验中系统预先给出了一版综合代码，将值传递、引用传递以及指针传递组合到两个被调函数中。例如在被调函数 swap 中涉及值传递和引用传递的组合，在被调函数 swapPointer 中涉及指针传递方式，学生也可以根据自己的需求自行修改代码，尝试不同的参数传递组合，点击可视化执行就能进入可视化界面。



图 12 扩展实验登录及代码编辑界面

步骤二：可视化界面分为两个大的板块。左边是代码部分，红色箭头表示即

将执行的代码，蓝色箭头表示上一步已执行的代码，通过点击上一步和下一步可以将代码向前或向后执行查看；右边是数据存储区域，变量存储的位置根据初始化方式的不同又可分为在栈中存储和在堆中存储，二者的区别在于栈中存放的是静态初始化的变量和函数，堆中存储的是用关键词 new 进行动态初始化的变量。可以看到 main 函数首先会被加载到栈顶，并在 main 中声明了 a, b 两个整型变量，此时 a, b 还未进行初始化赋值。

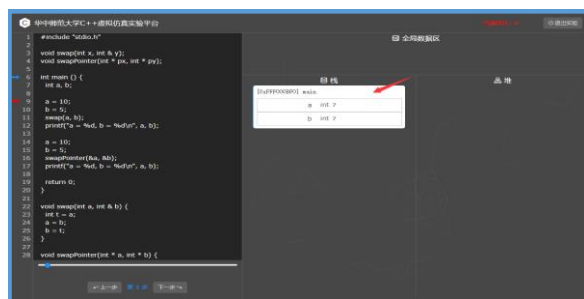


图 13 扩展实验界面

步骤三：继续点击下一步至代码第 11 行，main 中的变量 a, b 已完成赋值，在右边的数据区也能看到每个变量存储的数值，此时红色箭头指向 swap 函数，系统即将执行该处的代码。观察 swap 函数的形参可以发现，形参 a 使用的是普通的值传递，形参 b 使用的是引用传递，这将会给实参带来不同的影响。

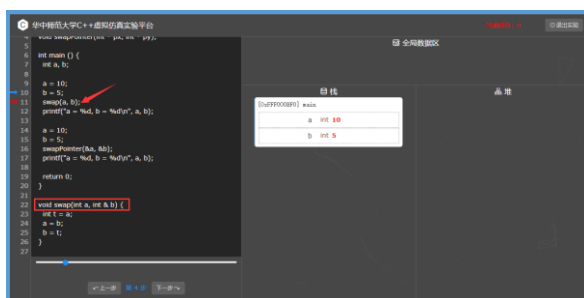


图 14 扩展实验执行过程

步骤四：继续点击下一步至代码第 22 行，swap 函数被加载到栈顶，其形参分别被初始化赋值，其中形参 a 使用实参 a 的拷贝进行赋值，形参 b 由于是引用传递，故形参 b 代表的仍然是实参 b 本身。在右边的数据区，我们使用箭头表示形参 b 指向了实参 b。

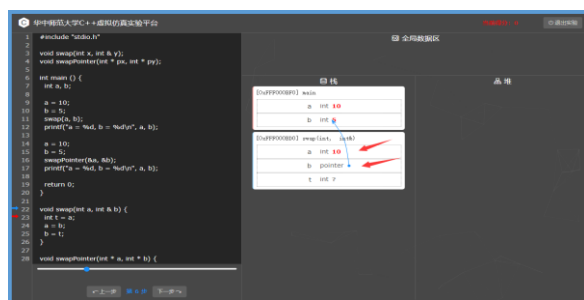


图 15 扩展实验执行过程

步骤五：继续点击下一步至代码第 25 行，swap 函数中定义了一个局部变量 t，并用形参 a 为其进行初始化赋值。蓝色箭头已执行的代码为将形参 b 的值赋值给形参 a，由于形参 b 代表的是实参 b 本身，故其值为 5，然后将 5 赋值给形参 a。注意形参 a 和实参 a 拥有不同的存储空间，它们之间不会相互影响，实参 a 的值不会发生变化，从数据区能清楚的看出这一点。

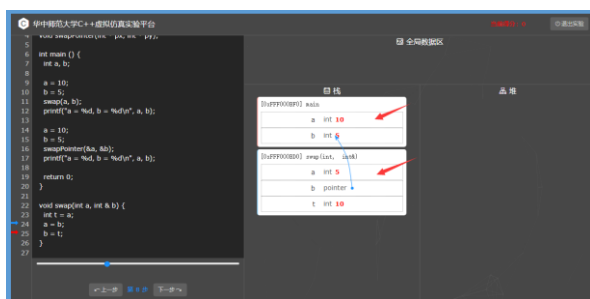


图 16 扩展实验执行过程

步骤六：继续点击下一步至代码第 26 行，蓝色箭头已执行的代码表示，将局部变量 t 的值赋值给形参 b，由于形参 b 代表了实参 b 本身，故实参 b 的值发生变化，从 5 变为了 10。

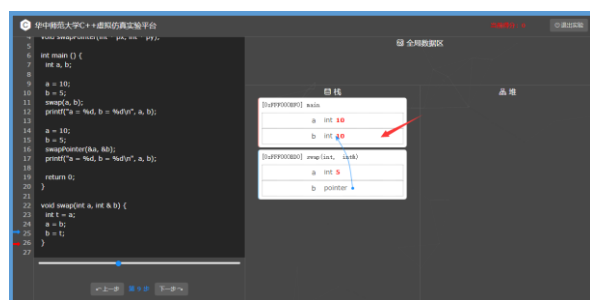


图 17 扩展实验执行过程

步骤七：继续点击下一步至代码第 7 行，swap 函数已执行结束，并从栈顶弹出，其中所有声明的局部变量及其所占空间都被系统回收。蓝色箭头所指代码表示输出 main 函数中变量 a, b 的值，学生需要在弹窗中输入自己的答案，以便核对结果是否正确。

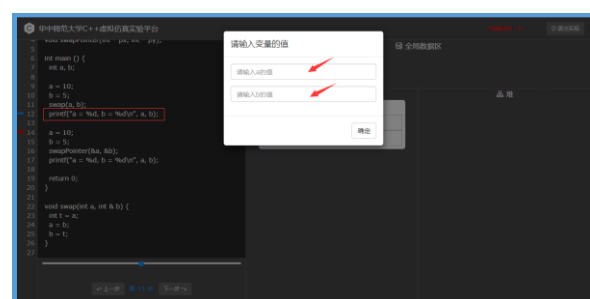


图 18 扩展实验执行时的问题交互

步骤八：继续点击下一步至代码第 28 行，main 函数中已对 a, b 变量重新赋值，数据区 a, b 的值也发生了变化。红色箭头指向 swapPointer 函数表示系统即

将执行该函数，可以看到该函数已被加载到栈顶，准备为其形参进行初始化赋值。

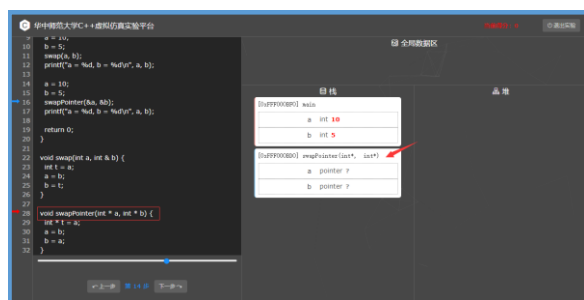


图 19 扩展实验执行过程

步骤九：继续点击下一步至代码第 30 行，swapPointer 中的形参都是指针变量，因此主函数中将实参 a，b 的地址传输进来，指针变量 a 接收了实参 a 的地址，并指向了 main 中的实参 a；指针变量 b 接收了实参 b 的地址，并指向了 main 中的实参 b。蓝色箭头已执行代码表示，在 swapPointer 中定义了一个指针变量 t，并用指针变量 a 为其初始化赋值，由于指着变量 a 保存了实参 a 的地址，并指向实参 a，赋值后指针变量 t 也保存了实参 a 的地址，并且指向实参 a。

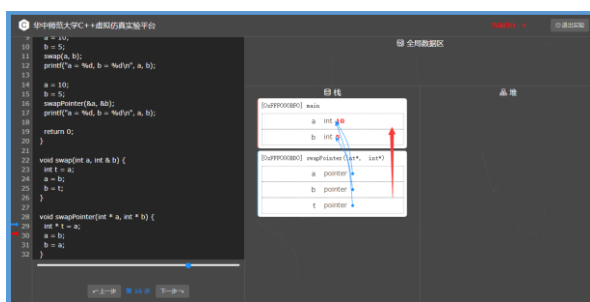


图 20 扩展实验执行过程

步骤十：继续点击下一步至代码第 31 行，蓝色箭头已执行代码表示，在 swapPointer 中将指针变量 b 赋值给指针变量 a，此时指针变量保存了实参 b 的地址，并且指向实参 b，注意这里只是进行地址的赋值，并不会改变原实参的值，数据区中有清晰的表示。

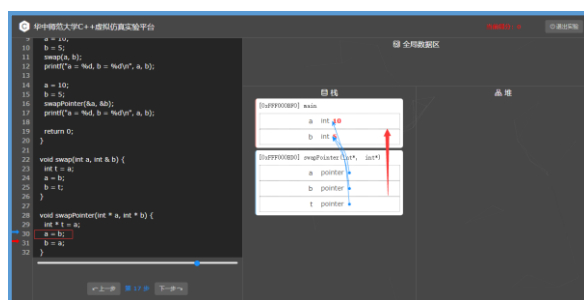


图 21 扩展实验执行过程

步骤十一：继续点击下一步至代码第 32 行，蓝色箭头已执行代码表示，在 swapPointer 中将指针变量 a 赋值给指针变量 b。由于指针变量 a 在上一步中已指向实参 b，再将指针变量 a 赋值给指针变量 b，指针变量 b 还是指向实参 b。注

意这里只是进行地址的赋值，并不会改变原实参的值，数据区中有清晰的表示。

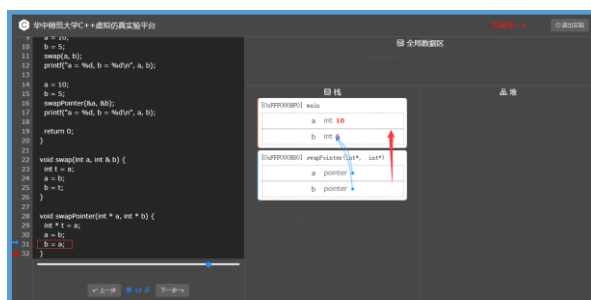


图 22 扩展实验执行过程

步骤十二：继续点击下一步至代码第 19 行，swapPointer 函数已执行结束，并从栈顶弹出，其中所有声明的局部变量及其所占空间都被系统回收。蓝色箭头所指代码表示输出 main 函数中变量 a, b 的值，学生需要在弹窗中输入自己的答案，以便核对结果是否正确。至此扩展实验部分完成，接下来进入理论测试部分。

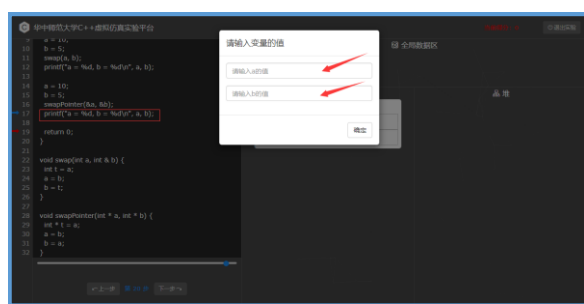


图 23 扩展实验执行时的问题交互

步骤十三：如果学生希望开展自定义扩展实验，可以退出扩展实验后重新进入扩展实验部分，并在线输入自己的 C++ 代码后点击“可视化执行”按钮开始实验。

理论测试部分

步骤一：点击理论测试进入理论测试部分，系统每次会从题库中随机抽取 10 道关于参数传递的问题，以选择题的形式提供给学生进行在线训练。



图 24 理论测试界面

步骤二：学生完成所有习题后点击最下方的提交测试即可完成本次在线测试，同时系统将给出每道题的正确答案供学生参考。为了更加深刻地理解和掌握每种参数传递方式的特点，学生可以选择回到基础实验部分或者扩展实验部分重复实

验，或者继续进行在线练习不断完善和巩固薄弱的知识点。



图 25 理论测试过程

3-7 实验结果与结论（说明在不同的实验条件和操作下可能产生的实验结果与结论）

（1）基础实验部分的实验结果与结论

实验条件 1：选择三个基础程序开展实验

结果与结论：是否完成了对应的基础实验程序，并正确的回答问题，根据每个基础实验的完成程度给与对应的分数。

实验条件 2：多次进行基础实验

结果与结论：允许多次进行基础实验。基础实验的最终成绩选择最高的基础实验成绩。

（2）扩展实验部分的实验结果与结论

实验条件 1：未开展或未完成扩展实验

结果与结论：扩展实验成绩按实际完成的程度计分。

实验条件 2：按系统推荐的扩展实验代码开展扩展实验

结果与结论：扩展实验成绩按实际完成的程度计分，主要步骤包括编译程序、调试程序、正确回答问题等。

实验条件 3：自定义扩展实验

结果与结论：自定义扩展实验除了系统评分外，具体实验难度由教师根据实验报告单独加分，评分依据包括程序代码难度、编译是否通过、执行过程中问题回答是否正确（注：本系统会自动根据自定义程序的输出设置问题）。

实验条件 4：多次进行扩展实验

结果与结论：允许多次进行扩展实验，自动以扩展实验记录在实验报告中，由教师评分。扩展实验的最终成绩选择最高的扩展实验成绩。

（3）理论测试部分的实验结果与结论

实验条件 1：多次进行理论测试

结果与结论：允许多次进行理论测试，每次随机从题库抽取 10 题进行考核，最终成绩选择最高的理论测试成绩。

实验条件 2：未完成或未提交理论测试

结果与结论：理论测试成绩记为 0 分。

(4) 最终实验成绩评分标准

系统会记录学生实验完成情况及步骤、结果，同时要求学生提交实验报告并由教师评分。最终实验成绩=系统评分的 40% + 实验报告评分*60%计算。

3-8 面向学生要求

(1) 专业与年级要求

专业：计算机科学与技术、软件工程、物联网工程、网络安全等相关专业；

年级要求：大学一年级、二年级。

(2) 基本知识和能力要求

基本知识要求：掌握计算机基本原理；掌握基本 C++程序设计语言；

能力要求：熟练使用浏览器；具有一定的 C++编程能力。

3-9 实验应用及共享情况

(1) 本校上线时间：2020 年 03 月 06 日（上传系统日志）

(2) 已服务过的学生人数：本校 285 人，外校 200 人

(3) 附所属课程教学计划或授课提纲并填写：

纳入教学计划的专业数：2

具体专业：计算机科学与技术、软件工程

教学周期：3，学习人数：285

(4) 是否面向社会提供服务：√是 ○否

(5) 社会开放时间：2020 年 05 月 01 日

(6) 已服务过的社会学习者人数：60 人

4. 实验教学特色

（该虚拟仿真实验教学课程的实验设计、教学方法、评价体系等方面的特色，限800字以内）

（1）实验设计特色

该实验最主要的设计特色在于将 C++ 代码执行过程进行了可视化。由于计算机系统的高度复杂性和不可见性，我们无法直观看到传统的代码编译执行过程，无法观察到每一个变量在执行过程中值的变化情况，也难以定位每个函数或变量在内存中存放的位置（栈或堆中）。虽然可以通过一些编译软件的 Debug 功能来单步执行代码，以查找每个变量的值，但 Debug 的过程对于入门者而言不容易掌握，而且也无法直观形象的可视化显示，甚至有时也难以发现代码中出现的逻辑错误。我们设计的 C++ 语言可视化执行虚拟仿真实验平台可以将代码的执行过程进行可视化，让程序在执行每一步代码的过程中，将变量的值及其存放的位置及时展示给用户。这样一旦可视化窗口中的变量值及其存放位置与用户的预期不一致，用户就能快速定位出错代码的位置。其次我们提供的可视化功能操作简单，只要认真阅读操作说明就能轻松上手，为 C++ 编程语言的学习者们带来极大的便利。

（2）教学方法特色

我们的教学方法遵从“开放性编程”、“理论与实践相结合”以及“循序渐进”的原则。

开放性编程：该虚拟仿真实验平台还支持用户自定义编程和可视化执行。用户可以根据自己对每个知识点的掌握情况，自行修改扩展实验中的示例代码，配合可视化功能，用户就能快速、轻松地实现自己的想法，不断开拓自己的思维。

理论与实践相结合：我们的实验分为基础实验、扩展实验和理论测试三个大的模块，前两个模块主要从代码层面出发，引导用户从代码编写与执行的实践操作中，了解不同参数传递方式各自的特点，熟练区分及使用不同的参数传递方式。理论测试模块则主要从理论知识层面出发，针对不同类型的参数传递规则，我们设计的每个问题都会从不同的角度进行考察，力求全方面地完善用户的理论知识体系，从根本上掌握每一个知识点。

循序渐进：在实验内容设置上，该实验既包含学生必须掌握的三种基础参数类型实验（按值传递、按引用传递以及按指针传递），也设计了两组扩展实验模块，将三种参数传递规则进行组合，设计出与真实的软件开发应用类似的场景，引导用户深入理解每种参数传递背后的机制。

（3）评价体系特色

该实验评价体系的特色主要体现在实验操作与理论测试相结合、基础知识的

客观评测与开放性实验的主观评价相结合。

实验操作与理论测试相结合：理论知识是实验操作背后的基础和本质，学生在实验操作过程中，应该带着理论知识去分析和理解实验过程、现象和结果，实验过程、现象和结果又可以帮助学生加深对理论知识的理解。因此，在该实验中，我们设置了 20 分的理论测试。如果学生对理论知识存在疑问，可以方便地重复实验来进行求证，从而达到实验操作与理论知识的相互促进。

基础知识的客观评测与开放性实验的主观评价相结合：在基础知识的评测方面，我们采用了程序化的客观评测，能够客观、快速、准确地评估用户的实验过程和结果。实验中的每一个关键步骤都设计了对应的考核机制，学生完成相应的步骤后就能自动获得相应分数。同时，该系统还提供了开放式的扩展实验，学生可以充分自由的发挥自己的想象力，交互式地设计编写 C++ 源程序并进行可视化运行，自由探索 C++ 的相关知识点及其执行时的内存状态。开放式的扩展实验以实验报告的形式提交，由教师根据实验内容评分。

5. 实验教学在线支持与服务

- (1) 教学指导资源: ☐ 教学指导书 ☒ 教学视频 ☒ 电子教材 ☐ 课程教案
(申报系统上传) ☒ 课件 (演示文稿) ☐ 其他
- (2) 实验指导资源: ☒ 实验指导书 ☒ 操作视频 ☐ 知识点课件库 ☒ 习题库
(申报系统上传) ☐ 测试卷 ☒ 考试系统 ☐ 其他
- (3) 在线教学支持方式: ☒ 热线电话 ☐ 实验系统即时通讯工具 ☐ 论坛
☒ 支持与服务群 ☐ 其他
- (4) 5 名提供在线教学服务的团队成员; 6 名提供在线技术支持的技术人员; 教学团队保证工作日期间提供 14 小时/日的在线服务。

6. 实验教学相关网络及安全要求描述

6-1 网络条件要求

- (1) 说明客户端到服务器的带宽要求 (需提供测试带宽服务)

带宽要求: 100KB/S 及以上, 正常上网速度即可满足;

理由: 本实验采用 Web 形式提供服务, 浏览器到服务器之间传输的内容只有网页、C++源代码和 JSON 格式的运行数据。交互式地可视化运行均以 Javascript 程序的形式在客户端执行, 操作步骤和成绩以 AJAX 请求提交, 对带宽要求很低。

- (2) 说明能够支持的同时在线人数 (需提供在线排队提示服务)

同时在线人数: 120 人

理由: 本实验采用了可扩展的构架体系, 服务端主要分两个模块: Web 服务端和后端。Web 服务端主要提供基础的网页和评分等服务, 属普通 Web 应用网站; 后端提供 C++源代码的编译、运行等服务, 是主要的压力来源, 但该后端只在扩展实验时被调用, 且可以自由地增加后端服务器以实现负载均衡。

6-2 用户操作系统要求 (如 Windows、Unix、IOS、Android 等)

- (1) 计算机操作系统和版本要求

无

- (2) 其他计算终端操作系统和版本要求

无

- (3) 支持移动端: ☐ 是 ☒ 否

<p>6-3 用户非操作系统软件配置要求（兼容至少 2 种及以上主流浏览器）</p> <p>(1) 非操作系统软件要求（支持 2 种及以上主流浏览器）</p> <p>√谷歌浏览器 <input type="checkbox"/>IE 浏览器 √360 浏览器 √火狐浏览器 <input type="checkbox"/>其他</p> <p>(2) 需要特定插件 <input type="radio"/>是 √否</p> <p>如勾选“是”，请填写：</p> <p>插件名称：（插件全称）</p> <p>插件容量：M</p> <p>下载链接：</p> <p>(3) 其他计算终端非操作系统软件配置要求（需说明是否可提供相关软件下载服务）</p> <p>浏览器分辨率：1920*1080 及以上；</p>
<p>6-4 用户硬件配置要求（如主频、内存、显存、存储容量等）</p> <p>(1) 计算机硬件配置要求</p> <p>CPU：主频 1.5G、4 核及以上；内存：4G 及以上；</p> <p>(2) 其他计算终端硬件配置要求</p> <p>无</p>
<p>6-5 用户特殊外置硬件要求（如可穿戴设备等）</p> <p>(1) 计算机特殊外置硬件要求</p> <p>无</p> <p>(2) 其他计算终端特殊外置硬件要求：√无 <input type="radio"/>有</p> <p>如勾选“有”，请填写其他计算终端特殊外置硬件要求：</p> <p>无</p>
<p>6-6 网络安全（实验系统要求完成国家信息安全等级二级认证）</p> <p>(1) 证书编号：</p> <p>收到最新通知后，已启动委托第三方公司测试工作，预计 7 月初可获得国家信息安全等级二级证书。</p> <p>(2) 请附信息系统安全等级保护备案证明</p> <p>正在办理中。</p>

7. 实验教学技术架构及主要研发技术

指标	内容
系统架构图及简要说明	<p>(1) 系统架构图</p> <pre>graph TD Browser[浏览器] --> NGINX[Web网关 Nginx] NGINX --> WebService[Web服务] NGINX -- JSON --> Backend[后端] subgraph WebServiceBox [Web服务] Website[Web网站] Tomcat[Tomcat] Website --- Tomcat end Tomcat --- MySQL[(MySQL数据库)] subgraph BackendBox [后端] Cplusplus[C++调试器] Valgrind[Valgrind动态分析器] Docker[Docker容器] Cplusplus --- Valgrind Valgrind --- Docker end BackendBox --- Linux[Linux服务器]</pre> <p>(2) 简要说明</p> <p>本实验平台采用 Web 方式开发，主要可以分为三个部分：后端、Web 服务端和前端。</p> <p>后端：采用 PHP 开发，运行在类 Linux 系统，作用是调用 Docker 容器为每个 C++ 程序动态地创建编译、运行环境，并采用 Valgrind 调试器获取程序运行的详细信息并以 JSON 格式返回给前端，避免潜在的安全隐患。</p> <p>Web 服务端：采用 J2EE 平台开发，运行在 Linux 或 Windows 环境，作用是控制和协调整个实验教学的运行。</p> <p>前端：采用 HTML5、JS、SVG 等技术开发，运行在浏览器上，作用是开展实验教学，与学生进行交互，从 Web 服务端和后端获取 JSON 数据，并可视化地展示程序运行的动态效果。</p>

实验 教学	开发技术	<input type="checkbox"/> VR <input type="checkbox"/> AR <input type="checkbox"/> MR <input type="checkbox"/> 3D 仿真 <input checked="" type="checkbox"/> 二维动画 <input checked="" type="checkbox"/> HTML5 <input type="checkbox"/> 其他
	开发工具	<input type="checkbox"/> Unity3D <input type="checkbox"/> 3D Studio Max <input type="checkbox"/> Maya <input type="checkbox"/> ZBrush <input type="checkbox"/> SketchUp <input type="checkbox"/> Adobe Flash <input type="checkbox"/> Unreal Development Kit <input type="checkbox"/> Animate CC <input type="checkbox"/> Blender <input type="checkbox"/> Visual Studio <input checked="" type="checkbox"/> 其他
	运行环境	服务器 CPU 16 核、内存 16GB、磁盘 1000 GB、 显存 0 GB、GPU 型号:无 操作系统 <input type="checkbox"/> Windows Server <input checked="" type="checkbox"/> Linux <input type="checkbox"/> 其他 具体版本: Ubuntu 20.04 数据库 <input checked="" type="checkbox"/> Mysql <input type="checkbox"/> SQL Server <input type="checkbox"/> Oracle <input type="checkbox"/> 其他 备注说明 (需要其他硬件设备或服务器数量 多于 1 台时请说明) 是否支持云渲染: <input type="radio"/> 是 <input checked="" type="radio"/> 否
	实验品质 (如: 单场景模型总面数、贴图分辨率、每帧渲染次数、动作反馈时间、显示刷新率、分辨率等)	分辨率: 1920*1080 及以上 实验原理: 按程序运行的逻辑, 动态可视化地展示计算机内存的分配情况。 动画: SVG 矢量动画 动作反馈时间: 0.1s 以内

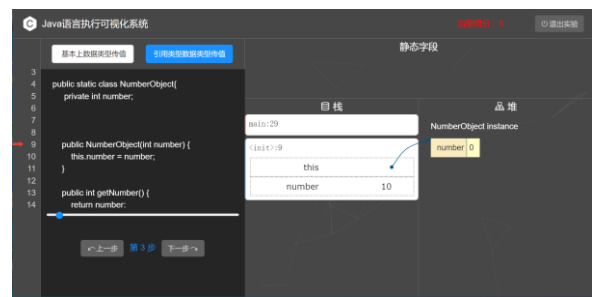
8. 实验教学课程持续建设服务计划

(本实验教学课程今后 5 年继续向高校和社会开放服务计划及预计服务人数)

(1) 课程持续建设

日期	描述
第一年	扩大实验教学规模，完善实验平台和服务
第二年	基于可视化执行平台，开发“C++类封装”实验
第三年	完善系统，为教师课堂教学、学生课外练习提供服务
第四年	基于可视化执行平台，开发“C++类继承”实验
第五年	基于可视化执行平台，开发“C++多态调用”实验

其他描述：教学组还将借鉴该可视化执行虚拟仿真实验平台的技术和经验，进一步开发 Java 语言可视化执行虚拟仿真实验平台，并以此为基础开发更多的 Java 语言可视化实验教学。目前，该 Java 可视化执行平台已开发出测试版，如下图所示。



(2) 面向高校、社会的教学推广应用计划

日期	推广高校数	应用人数	推广行业数	应用人数
第一年	3	240	2	60
第二年	5	400	4	120
第三年	5	400	4	120
第四年	5	400	4	120
第五年	5	400	4	120

其他描述：教学组不仅推广该实验教学，还将推广基于该平台的其他新增的实验教学项目。

9. 知识产权

软件著作权登记情况	
以下填写内容须与软件著作权登记一致	
软件名称	程序语言在线可视化仿真实验平台
是否与课程名称一致	<input type="radio"/> 是 <input checked="" type="radio"/> 否
<p>每栏只填写一个著作权人，并勾选该著作权人类型。如勾选“其他”需填写具体内容；如存在多个著作权人，可自行增加著作人填写栏进行填报。</p>	
著作权人	著作权人类型
华中师范大学	<input checked="" type="radio"/> 课程所属学校 <input type="radio"/> 企业 <input type="radio"/> 课程负责人 <input type="radio"/> 学校团队成员 <input type="radio"/> 企业人员 <input type="radio"/> 其他
权利范围	全部权利
软件著作登记号	2021SR0132405
如软件著作权正在申请过程中，尚未获得证书，请填写受理流水号。	
受理流水号	

10. 诚信承诺

<p>本团队承诺：申报课程的实验教学设计具有一定的原创性，课程所属学校对本实验课程内容（包括但不限于实验软件、操作系统、教学视频、教学课件、辅助参考资料、实验操作手册、实验案例、测验试题、实验报告、答疑、网页宣传图片文字等组成本实验课程的一切资源）享有著作权，保证所申报的课程或其任何一部分均不会侵犯任何第三方的合法权益。</p> <p style="text-align: right;">实验教学课程负责人（签字）：</p> <p style="text-align: right;">年 月 日</p>
--

11. 附件材料清单

1. 课程团队成员和课程内容政治审查意见（必须提供）

（申报课程高校党委负责对本校课程团队成员以及申报课程的内容进行政审，出具政审意见并加盖党委印章；团队成员涉及多校时，各校党委分别对本校人员出具意见；非高校成员由其所在单位党组织出具意见。团队成员政审意见内容包括政治表现、是否存在违法违纪记录、师德师风、学术不端、五年内是否出现过重大教学事故等问题；课程内容审查包括价值取向是否正确，对于我国政治制度以及党的理论、路线、方针、政策等理解和表述是否准确无误，对于国家主权、领土表述及标注是否准确，等等。）

2. 课程内容学术性评价意见（必须提供）

〔由学校学术性组织（校教指委或学术委员会等），或相关部门组织的相应学科专业领域专家（不少于3名）组成的学术审查小组，经一定程序评价后出具。须由学术性组织盖章或学术审查小组全部专家签字。无统一格式要求。〕

3. 校外评价意见（可选提供）

（评价意见作为课程有关学术水平、课程质量、应用效果等某一方面的佐证性材料或补充材料，可由课程应用高校或社会应用机构等出具。评价意见须经相关单位盖章，以1份为宜，不得超过2份。无统一格式要求。）