程序设计（II）：C++语言与面向对象程序设计 致谢信与复习要点

【致谢信】

各位同学：
大家好！
承蒙各位同学配合我的教学工作，《**程序设计（II）**》课程顺利结束。
我非常希望各位同学能顺利通过考试，愉快地回家过好暑假。

在 6 月 13~22 日之间，我每天会在东校区。需要答疑的同学，可提前发短信与我联系（我的手机：**13512768378**）。

由于端午节，下周二（2013-06-11）答疑改在本周日（2013-06-09）。下周三（2013-06-12）因端午节取消。

我都会按以下时间答疑：

1.周一（2013-06-09）	上午 8:00-11:30	学院楼 A308 答疑
	下午 14:30-18:00	
2.周二（2013-06-18）	上午 8:00-11:30	学院楼 A308 答疑
	下午 14:30-18:00	
3.周三（2013-06-19）	上午 8:00-12:25	教学楼 C304 或 C503 答疑
	下午 13:30-17:00	实验中心实验室答疑

我与二班同学共同学习了一个学期。在此，非常感谢各位同学配合我的工作，也使我在教学工作中找到自己知识结构的不足之处并督促我改进之。
所谓“教学相长”，我更希望在今后的工作中，能得到大家更多的意见、建议，帮助我改进教学中的缺点、改善教学方法。
本课程完成的 Projects，各班同学都付出了巨大的努力，在此一并感谢！

祝大家考试顺利，我更真诚地希望各位同学能在大学时光里学好、玩好，顺利地走上工作岗位！
祝大家过个开心的暑假，并将我的祝福带给你们的父母与家人！

万海
2013-06-05

【复习要点】

一、总论

1. 考试分为机试与笔试；
2. 机试有十道题，难度递增，完成越多且正确，得分越高（评价方式与往年有变化）；
3. 笔试题型；
4. 机试不得携带任何资料。

二、复习过程

1. 参考复习大纲或教学大纲；
2. 将课程讲义（PPT）仔细读一遍，考点不会超过 PPT 内容；关于教材；
3. 将本学期做过的实验重新完成一遍。

三、复习要点（着重学习带*，各要点权重我在课堂上已经提及，不再赘述）

1. **Introduction to C++**

   C++ programming style

   * from C to C++

   * C++ features.

2. **Classes and Objects**

   structures and classes;

   * classes definitions;

   * classes access control and class scope;

   classes, objects and abstract data types;

   * object creation, object initialization and object destruction;

   * default constructor, converted constructor, copy constructor;

   * encapsulation.

3. **Compile-Time Polymorphism: Function and Operator Overloading**

   function overloading;

   * operator overloading basics;

   * unary and binary operator overloading;

   * class members and friends:

   * two operator overloading approaches;

   * overloading examples of major operators;

   overloading and compile-time polymorphism.

4. **Run-Time Polymorphism: Inheritance and Dynamic Binding**

   * inheritance basics;

   * inheritance and multi-inheritance;

   * virtual base classes and virtual inheritance;

   * virtual functions, pure virtual functions, abstract base class;

   * static binding vs. dynamic binding;

   * run-time polymorphism;

   run-time type identification;

   design patterns basics.

5. **Templates, STL and Generic Programming**

   * templates:

* function templates and class templates;

* STL programming basics:

* containers, algorithms, iterators, adapters and allocators;

* generic programming and template, STL;

compile-time polymorphism and templates.

6. **Exception Handling**

* concepts of exception handling;

* user-defined exceptions;

* standard exception handling;

exception handling vs. error handling.

7. **I/O Streams**

* I/O stream class library;

* I/O stream basics: standard, file and string;

formatted I/O streams.

8. **Multi-paradigm Programming in C++**

evolution of programming methodologies: procedural programming, object-oriented programming and component-based programming;

* characteristics of C++ language: encapsulation, inheritance and polymorphism;

software reuse in C++: inheritance, polymorphism and templates;

cutting edge of software reuse: design patterns;

* generic programming in C++.

万海：whwanhai@163.com