《专业英语》作业 2-参考答案

英译汉如下英语材料，注意事项：

1）用 A4 纸或相当大小的纸完成作业，在作业纸的顶部写明“班级 姓名 学号”；

2）作业必须手写，不接收打印件与电子文件；

3）可借助工具查询单词，但勿采用工具翻译，尤其是图像识别翻译；

4）下次上课时由各班学习委员收集纸质作业，并按学号排序交给老师。

Object-Oriented Programming

Object-oriented programming (OOP) refers to a special type of programming that combines data structures with functions to create re-usable objects.

Otherwise, the term object-oriented is generally used to describe a system that deals primarily with different types of objects, and where you can take the actions depends on what type of object you are manipulating. For example, an object-oriented draw program might enable you to draw many types of objects, such as circles, rectangles, triangles, etc. Applying the same action to each of these objects, however, would produce different results. If the action is Make 3D, for instance, the result would be a sphere, box, and pyramid, respectively.

Many languages support object oriented programming. In OOP data and functions are grouped together in objects (encapsulation). An object is a particular instance of a class. [1] Each object can contain different data, but all objects belonging to a class have the same functions (called methods). So you could have a program with many e-mail objects, containing different messages, but they would all have the same functionality, fixed by the email class. Objects often restrict access to the data (data hiding).

Classes are a lot like types—the exact relationship between types and classes can be complicated and varies from language to language.

Via inheritance, hierarchies of objects can share and modify particular functions. You may have code in one class that describes the features all e-mails have (a sender and a date, for example) and then, in a sub-class for email containing pictures, add functions that display images. [2]Often in the program you will refer to an e-mail object as if it was the parent (super-class) because it will not matter whether the e-mail contains a picture, or sound, or just text. This code will not need to be altered when you add another sub-class of e-mail objects, containing (say) electronic cash.

参考译文：

面向对象程序设计

面向对象的程序设计涉及一种专用类型的程序设计，该技术将数据结构与函数相结合产生可重用的对象。

另一方面，术语“面向对象”通常用来描述一种系统，这种系统主要处理不同类型的对象，在这儿人们所采取的行为取决于他们正处理的对象的类型。例如，一种面向对象的绘图程序能使人们绘制多种类型的对象，如圆、长方形和三角形等。但是将同样的行为应用于不同的对象将产生不同的结果。例如，如果这一行为是制作三维，结果将分别形成球体、长方体和棱锥。

许多语言都支持面向对象的程序设计。在面向对象的程序设计中，数据和功能组合在一起归入对象内(封装)。一个对象是一个类的特例。每一个对象都可以包含不同的数据，但属于同一个类的所有对象具有相同的功能（称为方法）。于是，人们可以有一个带有许多电子邮件对象的程序，这些电子邮件对象包含有不同的信息，但具有电子邮件类固有的相同的功能。对象经常限制对数据的存储（数据隐藏）。

类很像类型——它们之间的关系会很复杂，而且随着语言的变化而变化。

通过继承，不同级别的对象可以共享和修改特殊的功能。可以在一个类中有代码描述所有邮件具有的特点（例如：发送者和日期），而在包含图片的子类中，增加显示图像的功能。在程序中人们经常查阅一个邮件对象，把它作为一个父类，因为该邮件是否包含图片、声音或仅仅一种文本无关紧要。当你添加另一个邮件对象的子类，譬如包含电子现金的时候不需要改变代码。