



7 - Task 7.1P

Object-Oriented Programming (Swinburne University of Technology)

OBJECT-ORIENTED PROGRAMMING

In modern technology, the number of data is increased dramatically. Whether it is in the video game, in robots or on the internet, the number of data inside those technologies is increased. It is very hard to code for those technology using traditional procedural programming. To manage those large about of data we need a new method to organise our code. Therefore, object-oriented programming is invented to manage complexity for larger software.

Object-oriented programming is basically grouping related data and functions together to form an object. An object has fields which know things and method which do things. An object is like a physical object. For example, a car is a physical object that has colour, shape, speed, fuel and etc and it also does things such as drive, lights on and lights off. So, the size, colour and shape of the car is field, and the driving is a method. Like a physical world, to build a car there should be a blueprint of car. The blueprint or template of an object is called a class. Using this class, we can create copies of objects as many as we want.

To design good software, we need to understand about classifications, roles, responsibilities and collaborations of objects. A role is a set of responsibilities and responsibility is a task which an object should do. So, an object may have many responsibilities, such as read, calculate, or print. And collaboration is the relationship between different classes. It is very important to identify clear collaboration and classification of objects. The good program should have high cohesion and low coupling. A cohesion means the methods and field inside the object should be related to the task. And the coupling is the linkage between objects. Lesser the number of links, the better the code is. Another most important concept is interface which helps to avoid overuse inheritance. And it also prevents from making errors. In order to become a master in object-oriented programming, we need to understand its four principles abstraction, encapsulation, inheritance and polymorphism.

Abstraction

In a simple word, abstraction is hiding the complexity of software. Abstraction is used in analyse and design state of programming. It helps to identify classifications, roles, responsibilities and collaborations of objects.

Encapsulation

Encapsulation is protecting data from the outsider. Encapsulation is used during the implementation and development of the program. Properties and access specifier are used to stop the outsider to access or change the data or the code. Encapsulation is used to implement the desired level of abstraction. Abstraction and encapsulation also make the codes high cohesive and low coupling.

Inheritance

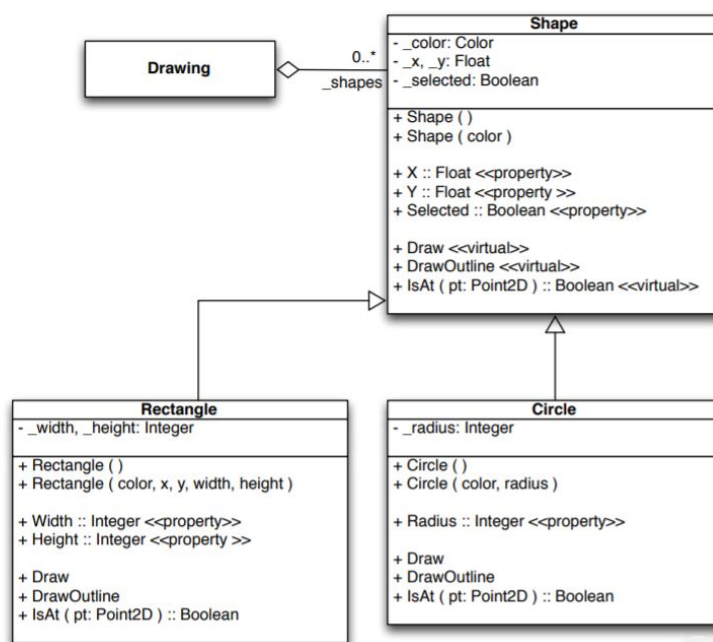
Inheritance is creating a class from an existing class. It helps programmers to save time write code and avoid complexity. It is a special type of relationship between classes, which is called is-a relationship. So, a child class is a type of parent/ base class.

Polymorphism

Polymorphism means having many forms or types. In OOP, polymorphism gives flexibility to the program. Polymorphism is used in many ways in programming. Using inheritance, also enable polymorphism in program such declaring an object as parent class can help to use any child classes.

In this subject, I have used all four principals in my programming.

In Drawing Program, I use abstraction and encapsulation by private the access of fields and by using properties. Using encapsulation stops the other object to access or change the data. Inherit rectangle, circle and line child classes from the parent class shape, helps to avoid duplication and save time to write code. Inheritance also creates polymorphism in my code. Such as instant of declaring an object like a rectangle or circle class, we can declare it as Shape Class which create flexibility.



Another example of OOP programming is The Maze Game. The maze game is console based game. All four principles are applied to this program. The name and description of the Game Object were assigned as private access to protect from outsiders. And the Item and Player classes were inherited from Game Object class, so it is easy to manage the code. Furthermore, the bag class was derived from the item class to create flexibility.

The object-orientated programming is very useful in modern technology to handle the larger code, since the amount of data, is used in technology, is increased day by day. And to

understand and use OOP successfully and effectively, we need to learn about four principles.