IT2154 Tutorial 3: OOP and Subtype Polymorphism

Question 1

What is encapsulation? Name one benefit of encapsulation. Support your argument with an example.

Encapsulation is like a way of grouping data together for example attributes or properties of a certain object or grouping methods which are functions that the object would have. The data would be grouped using class which is a one singular unit.

One benefit of encapsulation is its ability to hide data so that the data will be directly accessible globally/outside of object and instead can only be accessed using methods or functions known as accessors like getter methods and mutator such as setter methods.

For example I have a class called “Car” which encapsulates its properties such as the make, model, year and speed. So the attributes would be def \_\_init\_\_(self, model, make, year):

self.\_\_make = make

self.\_\_model= model

self.\_\_year = year

self.\_\_speed = 0

By doing the above, they should not be accessed directly from outside the class.

def get\_make(self):

Return self.\_\_make

def get\_model(self):

return self.\_\_model

These get methods are created to retrieve the properties of the car

Def acceleration(self):

Self.\_\_speed += 7

Def brake(self):

If self.\_\_speed >= 7:

self.\_\_speed -= 7

And these set methods are created to modify the function speed

Therefore encapsulating the data would only allow controlled access through these methods which will ensure that the data properties of the Car class would not be affected by any changes outside.

Question 2

What is abstraction via information hiding? Name one benefit of abstraction via information hiding.

Abstraction via information hiding is a concept in OOP where the data of the properties of a class are hidden globally. Only the data required will not be hidden.

A benefit of abstraction via information hiding is that the code will be easier to understand by developers.

Question 3

What is inheritance? Name one benefit of inheritance.

Inheritance is a concept in OOP where a child class inherits the properties and behaviors from its parent class. The child class would be able to access its parent class’s data and therefore use the attributes as well as methods of it, it would be able to define its own attributes and methods as well.

A benefit of inheritance is being able to reuse the code, as the child class can inherit the properties of its parent class, it can reuse the attributes and methods of its parent class.

Question 4

What is subtype polymorphism? Name one benefit of subtype polymorphism.

Subtype polymorphism is a concept in OOP where a base class can represent objects of its subclasses through a reference which would allow the substitution of derived class instances.

A benefit of subtype polymorphism is the ability to be flexible as you can write out codes that uses the objects of a base class without actually knowing the specific subclass of each object therefore new subclasses can be created without any changes to the current code that is using the base class.

# Question 5

Consider the following requirement, can you apply C# OOP concepts to design a mobile phone simulation system.

Each mobile has an IMEI code and a phone number. IMEI code is read-only, and phone number is readable and updateable.

There are two types of mobile. IOS and Android.

Android mobile stores a gmail account.

IOS mobile stores an apple ID.

Each mobile has an install application feature.

When installing an application

Android mobile prints out "communicating with Google Play using the\_gmail\_account". IOS mobile prints out "communicating with Apple App Store using the\_apple\_id."

Add a Main method to show case the subtype polymoprhism.

using System;  
using System.Collections.Generic;  
namespace MobileSimulation  
{  
public abstract class Mobile  
{  
public int IMEI { get; }  
public int PhoneNumnber { get; set; }  
public Mobile( int IMEI, int PhoneNumber )  
{  
this.IMEI = IMEI;  
this.PhoneNumber = PhoneNumber;  
}  
public virtual void InstallApp() { };  
}  
public class IOS : Mobile  
{  
public string AppleID { get; set; }  
public IOS ( int IMEI, int PhoneNumber, string AppleID )  
: base ( IMEI, PhoneNumber) { this.AppleID = AppleID; }  
public override void InstallApp()  
{  
Console.WriteLine($"communicating with Apple App Store using {this.AppleID}.");  
}  
}  
public class Android : Mobile  
{  
public string GMailAccount { get; set; }  
public Android ( int IMEI, int PhoneNumber, string GmailAccount )  
: base ( IMEI, PhoneNumber) { this.GMailAccount = GMailAccount; }  
public override void InstallApp()  
{  
Console.WriteLine($"communicating with Google Play using {this.GMailAccount}.");  
}  
}  
public static void Main(string [] args)

List<Mobile> mobiles = new List<Mobile>();  
IOS ios = new IOS(990000862471854, 90123456, "smartboy@apple.com");  
Android android = new Android (351756051523999, 99876543, "cool.lady@gmail.com");  
mobiles.Add(ios);  
mobiles.Add(android);  
foreach (var mobile in mobiles)  
{  
mobile.InstallApp();  
}  
}  
}  
4

1 / 1