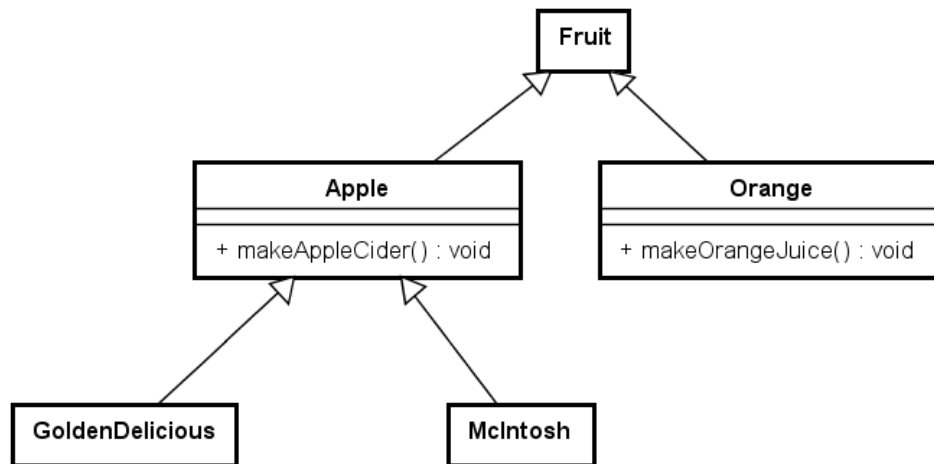


## A1 (25 marks)

*Focus: OOP basics, Inheritance, and Polymorphism*

Q1. [15 marks] Suppose that Fruit, Apple, Orange, GoldenDelicious, and McIntosh are defined in the following inheritance hierarchy:



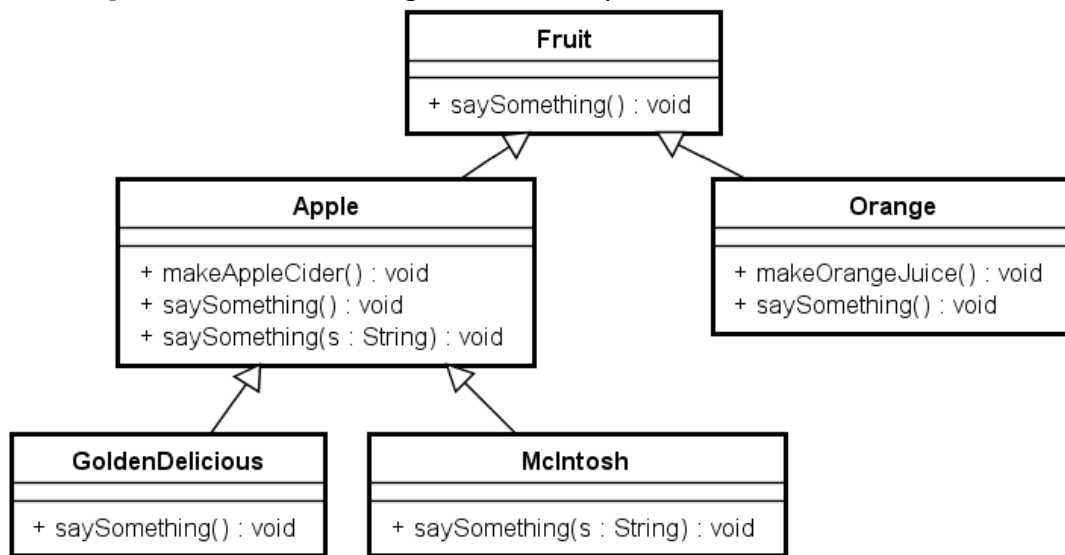
And assume that the following code is given:

```
Fruit fruit = new GoldenDelicious();  
Orange orange = new Orange();
```

Answer the following questions, and explain your answer.

- Is `fruit instanceof Fruit`?
- Is `fruit instanceof Orange`?
- Is `fruit instanceof Apple`?
- Is `fruit instanceof GoldenDelicious`?
- Is `fruit instanceof McIntosh`?
- Is `orange instanceof Orange`?
- Is `orange instanceof Fruit`?
- Is `orange instanceof Apple`?
- Can `fruit` invoke `makeAppleCider` method?
- Can `orange` invoke `makeAppleCider` method?
- Can `fruit` invoke `makeOrangeJuice` method?
- Can `orange` invoke `makeOrangeJuice` method?
- Is the statement `Orange p = new Apple()` legal?
- Is the statement `McIntosh p = new Apple()` legal?
- Is the statement `Apple p = new McIntosh()` legal?

Q2. [10 marks] Consider the following class hierarchy.



A method `saySomething()` displays the type of object, e.g., for the Apple class, the method displays “I am an Apple”. The overloaded method `saySomething(String s)` displays the type then the value of `s`, e.g., for class Apple and `s = “hello”`, the method displays “Apple says: hello!”

a) What is displayed on the screen after running the code below.

```
public class A2 {
    public static void main(String[] args) {
        Fruit[] fruits = new Fruit[5];
        fruits[0] = new Fruit();
        fruits[1] = new Apple();
        fruits[2] = new Orange();
        fruits[3] = new GoldenDelicious();
        fruits[4] = new McIntosh();
        for(Fruit f1: fruits){
            f1.saySomething();
        }
        Fruit[] apples = new Apple[3];
        apples[0] = new Apple();
        apples[1] = new GoldenDelicious();
        apples[2] = new McIntosh();
        for(Fruit f2: apples){
            //add code here
        }
    }
}
```

b) Write the missing code where indicated above in order to invoke the method `saySomething(String s)` using the reference variable `f2`. Use any value for `s`. Explain the output (i.e., why we see each text line in the output). *Hint: object casting.*

c) Is the following statement valid? Explain.

```
apples[0] = new Orange();
```

---

**Note about the array `f2` in Q2 above:**

- The array is of the type `Fruit`. However, each element is of the type `Apple`. This means:
    - o the array elements can only be of the type `Apple` or a subtype of `Apple`.
    - o The reference `f2` can only invoke methods known to `Fruit` (i.e. it cannot for example invoke methods in the `Apple` class if they are not defined in `Fruit`)
  - We do this (array type is different from elements type) when we want to limit the programmer to use only the methods in the array type (e.g. `Fruit` in our example) and put a restriction on the type of objects that can be stored in the array (e.g. only `Apples` can be stored in `f2`. We cannot store oranges in the array).
- 

## Submission Instructions

For this assignment, you need to do the following:

- 1- Create a new Microsoft Word file which name consists of **your student number followed by the assignment number**, e.g., "1234567\_assignment\_A1.docx".
- 2- Write your answers in the above file and submit it to **Canvas**.

Note that you can resubmit an assignment, but the new submission overwrites the old submission and receives a new timestamp.