**CS2030S**

**RECITATION 3**

**Q1**

a)

i) No, as Shape does not implement Printable thus from the run-time type of a Shape object, it does not have the methods from a Printable interface

ii) Printable is an interface so it should be able to call methods

iii) calling getArea() from the runtime type of Shape, unless the getArea() has been overridden, should be fine?

iv)Printable does not have the method getArea() defined it it, so it will result in an error

b)No, as an object (in this case, Circle) can only extend from one class.

c) assuming interfaces can extend each other, yes they could work

**Q2**

There could be security implications as the child of multiple inheritance could access one of the parent class’s method whose intended purpose is in direct conflict with the other parent class’s intended purpose.

Eg one of the parent class A has a setter method while the other parent class B does not. Parent B’s omission of the setter method is such that the child classes cannot set their attributes but Parent A’s setter method enables the child class to set its attributes

**Q3**

Yes, as the toggleUnderline() in PlainText should return an underlined Text (ie FormattedText object) yet when it returns this, it returns a PlainText object (ie no underlining)

**Q4**

It complies. But if we swap the methods with the exact body, then it does not compile. Reason being B is a subclass of A, which means A cannot return B. But B can return an A object since it is the subclass of A.