

3 Object-Oriented Programming (acr31)

You have been asked to implement a user-interface component `ButtonCanvas` which is both a button that can be clicked by the user and a canvas which the user can draw on.

Your existing code contains a `Button` class and a `Canvas` class. Both of these extend `GuiComponent`.

- (a) `ButtonCanvas` could be built using *multiple inheritance*. What does this mean in this context? [2 marks]
- (b) Give two reasons why this might be desirable. [2 marks]
- (c) Give two complexities that arise in this case. [2 marks]
- (d) Java interfaces originally contained only *abstract methods* and *static final* fields. How did this restriction avoid the complexities of extending multiple classes? [3 marks]
- (e) Draw a UML diagram for building `ButtonCanvas` using abstract interfaces rather than multiple inheritance. Explain how your design attempts to preserve the desirable properties arising from multiple inheritance.

You do not need to recall the exact UML specification, instead you may provide a key explaining your notation. [9 marks]

- (f) Recent versions of Java added default methods to interfaces. What is the impact of this with respect to multiple inheritance? [2 marks]