4 Object-Oriented Programming (RKH)

(*a*) Using example Java code, distinguish between overloading, overriding and shadowing when applied to Java methods. [6 marks]

(*b*) For each of the numbered lines of code below, state whether it will compile. If it does, state and explain the output it gives when run. If it does not, explain why not. [10 marks]

```
public static class A {
    public void print() {System.out.println("A");}
    public void printObject(A a) {System.out.println("A");}
    public static void printAll(List<A> list) {
        for (A a : list) a.print();
    }
}

public static class B extends A {
    public void print() {System.out.println("B");}
    public void printObject(B b) {System.out.println("B");}

    public static void main(String[] args) {
        A a = new A();
        B b = new B();
        LinkedList<A> alist = new LinkedList<A>();
        alist.add(a);
        alist.add(b);
        LinkedList<B> blist = new LinkedList<B>();
        blist.add(b);

1       ((B)a).print();
2       ((A)b).print();
3       A.printAll(alist);
4       b.printAll(blist);
5       a.printObject(b);
6       b.printObject((B)a);
7       b.printObject((A)b);
    }
}
```

(*c*) All methods in Java are dynamic polymorphic as a design choice. Discuss the advantages and disadvantages of making dynamic polymorphism *optional* for Java methods. [4 marks]