3 Object-Oriented Programming (RKH)

An online retailer uses custom Java software to manage their inventory and sales.

(*a*) Each product sold is represented using an immutable `Product` object. Explain what is meant by immutable, how immutability is typically achieved in Java and the advantages of using immutable objects in general. [4 marks]

(*b*) `Product` objects are requested through a `Product getProduct(long code)` method, which returns a reference to a `Product` object given a valid product code. Product information is often re-requested as customers make their selections so the 10,000 most recently *accessed* `Product` objects are cached in memory. Uncached `Product` objects are created with information retrieved from a database when requested.

(*i*) The cache uses a `java.util.HashMap` and a custom implementation of a doubly-linked list. The list keeps an ordering over the `Product` objects where more recently used objects are at the front. The `HashMap` provides fast lookup into the list. Show that this scheme gives a constant (O(1)) running cost for `getProduct()`, ignoring the cost of the database lookup. [3 marks]

(*ii*) Create a class `Store` that implements the cache as described. You need only define `getProduct` and any state or definitions it needs. All other state and methods can be ignored. You may assume the existence of a method `loadFromDatabase(long code)` that will create a `Product` object for product code `code` or return `null` if the code is invalid, and that a `Product` object has a `long getProductCode()` method that returns its product code. [10 marks]

(*c*) A customer's basket of items, represented by a class `Basket`, can be viewed as a list of products. Therefore `Basket` might extend `LinkedList<Product>`. Compare this approach to a `Basket` that contains ("has-a") `LinkedList<Product>` instead. [3 marks]