International study centre

Linked List and Package Diagrams

❑Name : John Alamina

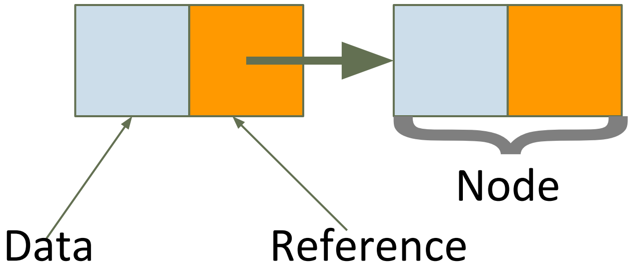
❑Email : john.alamina@hud.ac.uk

Contents

- ❑Vector vs Linked List
- ❑Linked List ADT
- ❑Package Diagram
- ❑Event-Response Modelling

Limitations on Arrays and Vectors

- The major limitation with arrays and vectors is the need to occupy contiguous blocks of memory and a more efficient way to manage memory is using a linked list.
- A linked list is an advanced data structure like a a dynamic array that grows by each element having a reference to the next element in the sequence.



Linked List ADT

- A Linked list has two components the data and the reference.
- Operations on Linked list includes
    a. Adding to the list
    b. Traversing the list. That is printing the list
    c. Removing from the list
    d. Finding data in the list

Linked List ADT

- The procedural Linked List interface may look like this

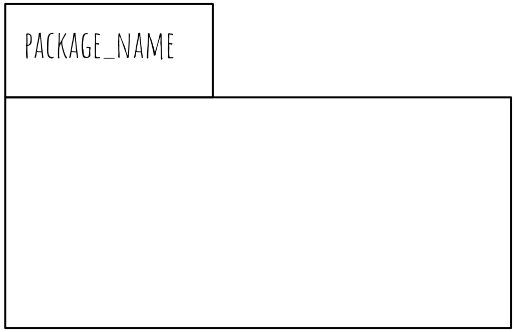
```
struct node{
int data;
struct node *next;
};
node *start=NULL;
node *createNode();
void insertNode();
void printlist();
boolean exists(int data);
void deleteNode();
```

Exercise

- Write an object-oriented version of the Linked List interface given in the previous slide.

Package Diagram

- Package Diagrams are used to organise a set of classes or other UML components into logical units.
- The package diagram symbol is given in the illustration below. It consists of a rectangle with a tab, making it look like a folder containing files.

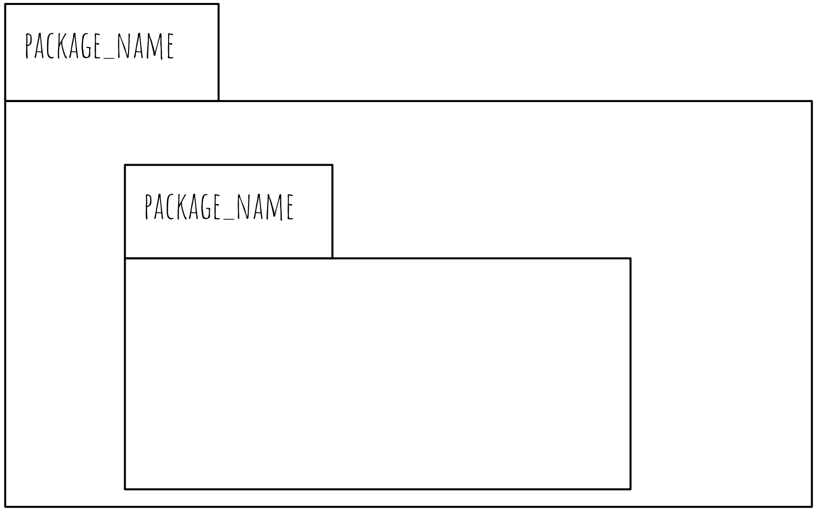


Package diagram

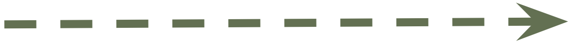
- Package diagram also include symbols for dependency and generalisation
- A package diagram classes or objects may also be made public and private depending on whether they can be accessed outside the package or not.
- Note that packages can be nested within other packages and the static member of (::) operator can be used to show nested packages of classes

Package diagram

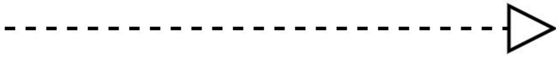
- In C++ packages are implemented as namespaces



Package Nesting



Dependence



Generalisation

Event Response Modelling

Event response modelling refers to how applications respond to events within the software system.

A common event-response model is handling user input.

Two common strategies for event-response modelling includes

1. Event loop (procedural)
2. Observer (object-oriented model)

We will consider a simple procedural model with and without a loop structure.

Exercise

Write a simple event-response model for the different forms (login, register and refresh password) of the authentication manager system.

Any Questions?