

ENGINEERING TRIPOS PART IIA

---

Day Date April 2010 9 to 10.30

---

Module 3F6

SOFTWARE ENGINEERING AND DESIGN

CRIB

*Answer not more than three questions.*

*All questions carry the same number of marks.*

*The approximate percentage of marks allocated to each part of a question is indicated in the right margin.*

*There are no attachments.*

STATIONERY REQUIREMENTS

Single-sided script paper

SPECIAL REQUIREMENTS

Engineering Data Book

CUED approved calculator allowed

<p>You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator</p>
--

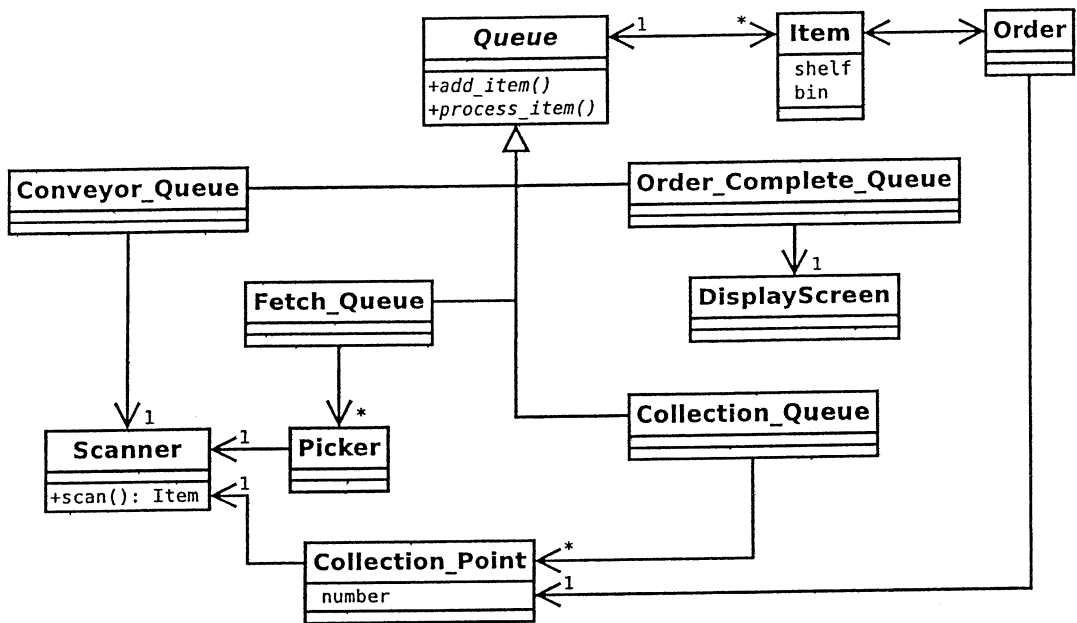
- 1 (a) Why are *data hiding* and *encapsulation* benefits of object-oriented design? [10%]

Data hiding keeps the implementation details of a class hidden from the rest of the application. This makes it easier to maintain invariants and to change the data representation should that be desired. Encapsulation adds functions to each class so that they can be responsible for managing their own data.

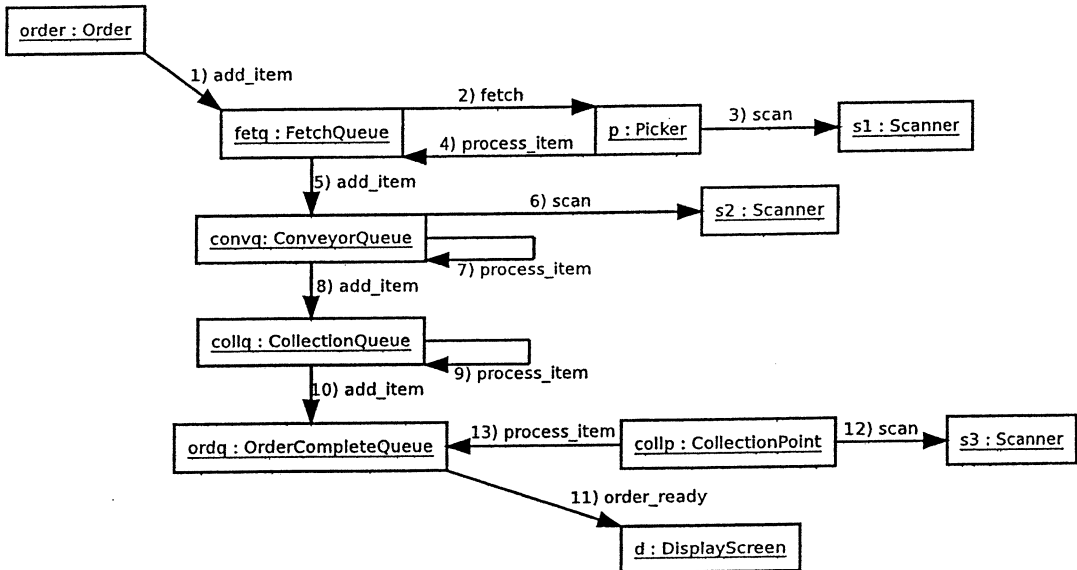
- (b) A high street shop operates according to the procedure shown in the box below:

1. A Customer places an order for items at a purchase counter and pays for them. The customer is given a receipt for the goods which contains their order number and then waits for their items to be collected for them.
2. Items for the orders are collected by “pickers” in the back of the shop. Each picker has a computer terminal which displays the description, shelf number and bin number of an item for them to collect. They collect that item from the shelves, scan it with the barcode scanner attached to their computer terminal and then place the item on a conveyor belt. They are then given a new item to collect.
3. The conveyor belt delivers the items to a distribution area in the front of the shop. Staff there scan each item as it arrives with a barcode scanner. The computer terminal attached to the scanner then displays the order number that it belongs to and the number of a collection counter. The staff member then takes the item to that collection counter.
4. When an order is complete, the customer is informed of this via a large display screen that indicates which collection counter they should go to.
5. The customer goes to this collection counter and a member of staff scans their receipt with a scanner. This brings up a list of goods on the display for that collection point. Each item is then scanned as it is handed over to the customer who takes the items and leaves the store.

- Using good design principles, draw a UML class diagram that shows the main classes that will be needed for the software to run this system. [50%]



(c) Draw a collaboration diagram that shows the sequence of events when a customer purchases one item. [40%]



2 (a) Describe the sequence of events that takes place when a remote call is made using CORBA. [30%]

1. A program makes a call on a local proxy
2. the proxy contacts the local ORB and passes the call arguments to it
3. the local ORB contacts the remote ORB and passes the call arguments
4. the remote ORB makes the call on the remote servant
5. the remote servant returns
6. the remote ORB sends the return value to the local ORB
7. the local ORB returns the value to the local proxy
8. the local proxy returns the value to the calling program

(b) It is desired to extend the functionality of some spreadsheet software by allowing cells within spreadsheets to reference cells in other spreadsheets over a network using CORBA.

To do this, each spreadsheet is given a name. A central server is set up and each spreadsheet registers its name with that server. A cell within a spreadsheet can then contain a special formula which identifies the name of a spreadsheet and the row and column of a cell within that spreadsheet. When the formula is entered, the central server is used to obtain a reference to the relevant spreadsheet and then, whenever the formula is evaluated, a call is made to that spreadsheet to obtain the current value of that cell. Give the CORBA IDL that will be needed to enable this.

[30%]

```
module NetworkSpreadsheet {  
    interface spreadsheet {  
        float get_cell(in long row, in long col);  
    }  
}
```

```

}

interface server {
    void register (in spreadsheet s, in string name);
    spreadsheet get_spreadsheet(in string name);
}
}

```

(c) Why might this approach generate a lot of network traffic? How could the design be improved so as to reduce the amount of network traffic? Give any CORBA IDL necessary for the new design. [40%]

This would cause a lot of traffic because the formula is typically updated much more often than the value of the cell referenced. It would be much better for the referenced spreadsheet to inform other spreadsheets when the cell is updated using the observer pattern:

```

module NetworkSpreadsheet {
    interface CellObserver {
        void notify (in float value);
    }

    interface spreadsheet {
        void observe_cell(in long row, in long col, in CellObserver obs);
    }

    interface server {
        void register (in spreadsheet s, in string name);
        spreadsheet get_spreadsheet(in string name);
    }
}

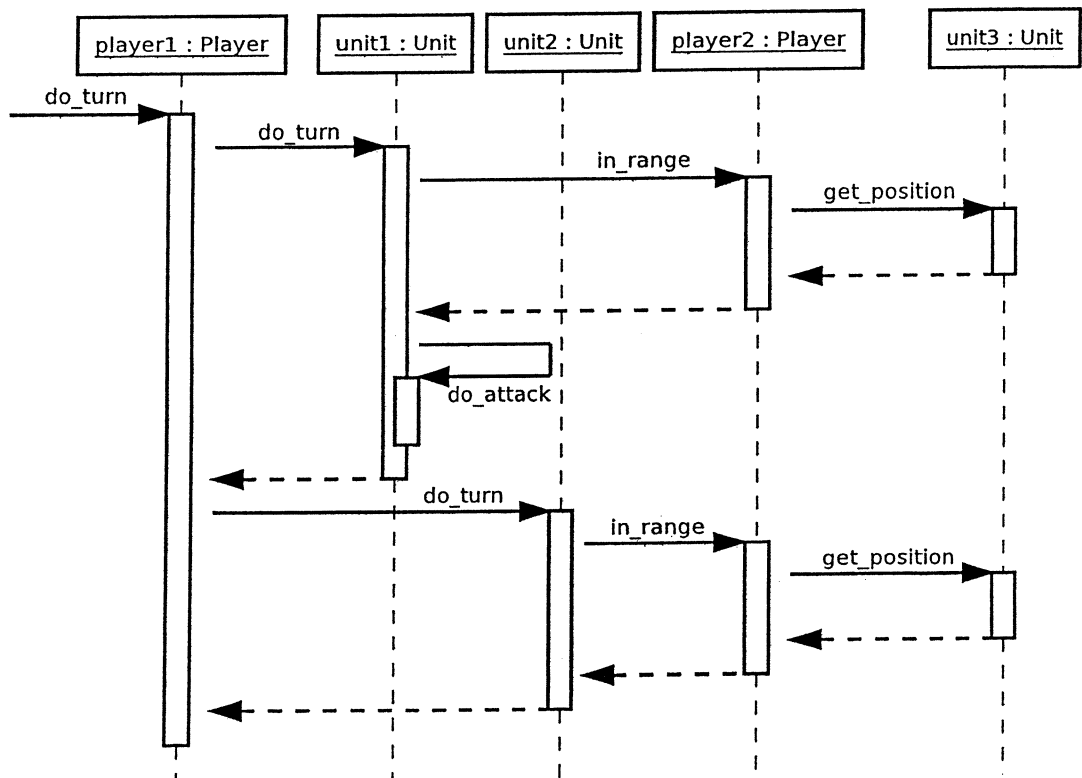
```

(TURN OVER

- 3 (a) What is meant by polymorphism in object-oriented design? [10%]

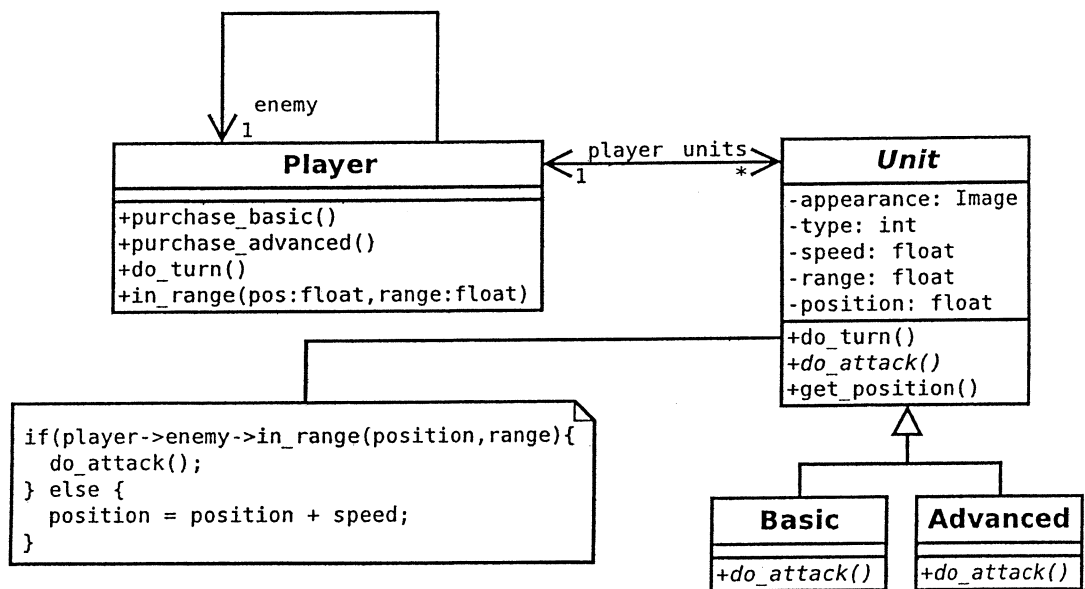
Polymorphism refers to the concept of virtual functions. A virtual function allows the type of an object to be identified at run-time and the function appropriate for that object to be called. For example a rectangle and an ellipse might both be derived from Shape with a virtual draw function. When draw is called on a shape, the type of shape is determined (rectangle or ellipse) and the appropriate draw function for that shape is called. This allows the production of heterogeneous containers containing both rectangles and ellipses.

- (b) Figure 1 shows a UML class diagram for part of a computer game for two players. The object diagram showing the state of a particular game is shown in Figure 2. Draw a sequence diagram which shows what happens when the do\_turn() function is called for player1. [40%]

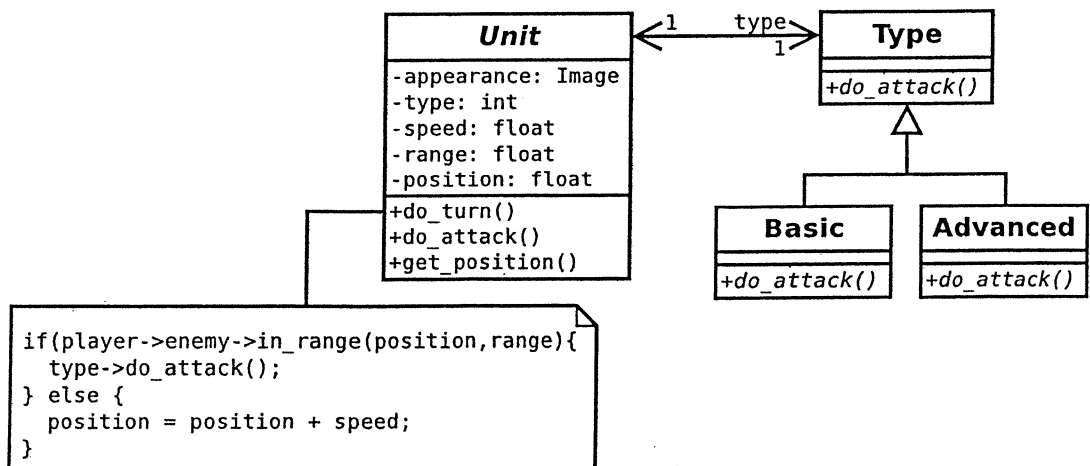


(c) Why is the use of the type attribute in class Unit a bad idea? Draw a class diagram for an improved design. You do not need to show how any functions are implemented. [15%]

Typecodes are a bad design in general because polymorphism can produce the same functionality more elegantly. In particular, code that uses type codes can be harder to extend than code that uses polymorphism. A more elegant design would be:

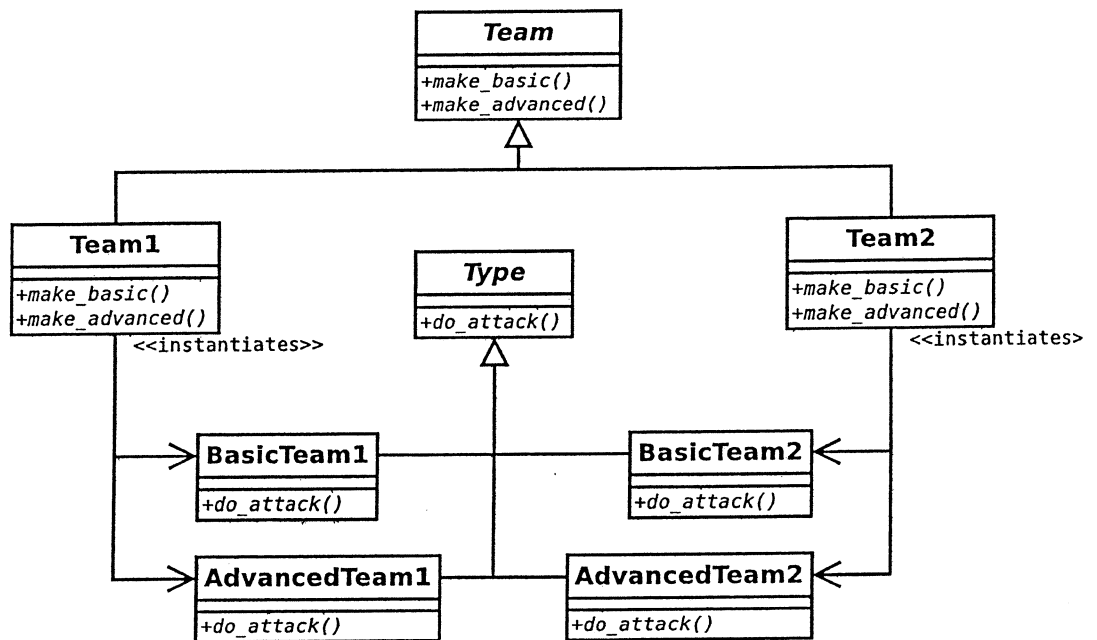


(d) Now suppose it is necessary to allow units to be upgraded from basic to advanced during the game. How should the design be modified to allow this? Give your answer as a class diagram. [10%]



(e) It is now desired to support multiple teams. Each team has its own basic and advanced units which take different actions when the `do_attack()` or `do_special_attack()` functions are called. When the game begins the player chooses a team and can then only create units belonging to that team. Draw a class diagram showing the design needed to achieve this functionality. [25%]

This is easiest done using the factory design pattern:





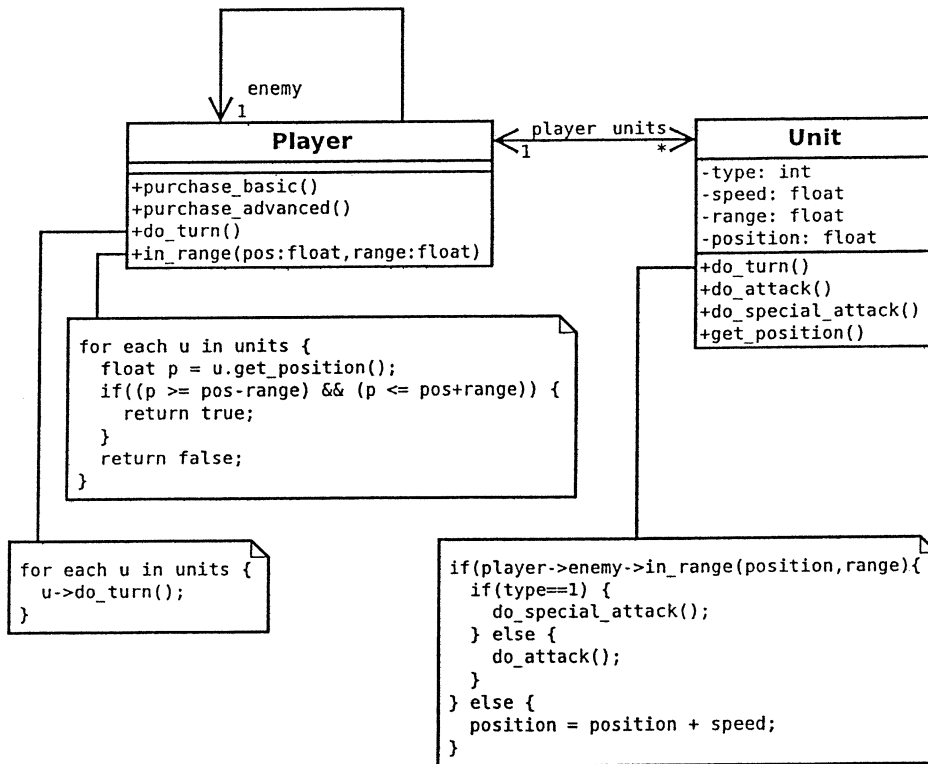


Fig. 1

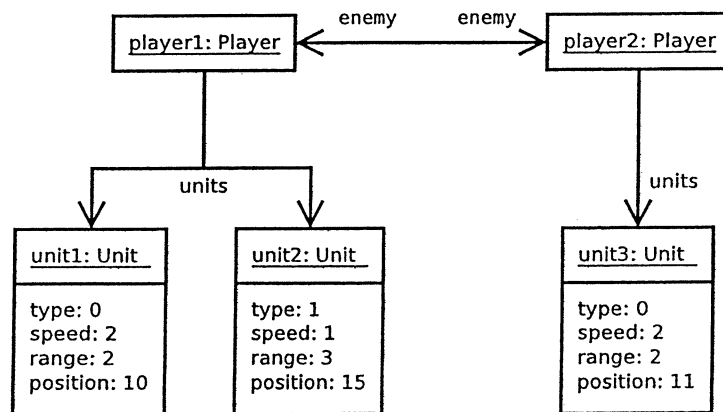


Fig. 2

(TURN OVER)

- 4 (a) What does the term *thread-safe* mean? [20%]

A thread-safe function can be called from multiple threads concurrently without interfering with its operation. In particular, this means that critical sections must be protected by mutexes or semaphores.

(b) A multi-player online computer game in which players have bank accounts that store funds which can be used to purchase in-game items or trade with each other is implemented using the following class:

```
class Account {
public:
    bool transfer_to(Account& ac_to, int amount) {
        if(balance < amount){
            return false;    // insufficient funds
        }
        balance = balance - amount;
        ac_to.balance = ac_to.balance + amount;
        return true;
    }

private:
    int balance;
};
```

Why is the definition of `transfer_to()` not thread-safe? Give an example of how this function might fail to operate as intended when called from multiple threads concurrently. [20%]

The lines

```
balance = balance - amount;
ac_to.balance = ac_to.balance + amount;
```

are not threadsafe because balance may be altered in one thread between being read and written to in another. For example if two transfers are being made from the same account in two different threads, balance may be read in both threads concurrently, the amount subtracted from that balance in each thread and then the results written back one after the other, with the second result overwriting the first.

(c) Give a modified version of the Account class that uses Semaphores to make the code thread-safe. Why might deadlocks be an issue? Suggest how they might be avoided. [40%]

```
class Account {
public:
    bool transfer_to(Account& ac_to, int amount) {
        sem.enter();
        if(balance < amount){
            return false;    // insufficient funds
        }
        ac_to.sem.enter();
        balance = balance - amount;
        ac_to.balance = ac_to.balance + amount;
        ac_to.sem.leave();
        sem.leave();
        return true;
    }

private:
    Semaphore sem;
    int balance;
};
```

This can cause a deadlock if two transfers are taking place in opposite directions simultaneously if each thread acquires the lock on its own sem and is waiting for the lock on ac\_to.sem. This could be avoided by numbering the accounts and obtaining the locks on the two accounts in order. Thus:

```

class Account {
public:
    bool transfer_to(Account& ac_to, int amount) {
        if(ac_num < ac_to.ac_num){
            sem.enter();
            ac_to.sem.enter();
        } else {
            ac_to.sem.enter();
            sem.enter();
        }
        if(balance < amount){
            sem.leave();
            ac_to.sem.leave();
            return false;    // insufficient funds
        }
        balance = balance - amount;
        ac_to.balance = ac_to.balance + amount;
        sem.leave();
        ac_to.sem.leave();
        return true;
    }

private:
    Semaphore sem;
    int balance;
    int ac_num;
};

```

(d) If there is one account that is heavily used (for example the account belonging to the in-game bank), why might the code not run efficiently? Suggest methods that might make this more efficient. [20%]

If one account is heavily used, its lock will only allow one thread to access it at a time, leaving many threads waiting to obtain the lock. One solution to this would be to create multiple trading accounts for the bank, so that these can be used

independently in separate transactions.

END OF PAPER