

PRIFYSGOL CYMRU; UNIVERSITY OF WALES

DEGREE EXAMINATIONS JANUARY 2003

SWANSEA

Computer Science

CS 214 Object Technology

Attempt 2 questions out of 3

Time allowed: 2 hours

Students are permitted to use the dictionaries provided by the University

Students are NOT permitted to use calculators

CS_214 OBJECT TECHNOLOGY

(Attempt 2 questions out of 3)

Question 1.

- (a) Consider a class **Customer** which has features of name and title (both of which are **Strings**). Write a constructor for this class which takes two **String** parameters and sets the features accordingly.

[3 marks]

- (b) Consider a class **Stack** of **Customers** which contains the following methods and explanatory comments:

```
void push(Customer c); //Pushes customer c onto the top of the stack
Customer top(); //Returns the customer at the top of the stack
Boolean empty(); //Is the stack empty?
void pop(); //Removes the top customer from the stack
```

Complete the following code segment which may appear in a client of the class Stack. The code should read the value of **i** and perform the appropriate action :

```
void takeAction(int i){
    PrintWriter screen = new PrintWriter(System.out, true);
    Stack myStack = new Stack();
    String inTitle = "Mr";
    String inName = "Smith";
    /*if i ==
        1. Create a new Customer with the values of inName and inTitle
           (using the above constructor) and add to the stack.
        2. If the stack is empty print an error message to the screen,
           otherwise remove a customer from the stack.
        3. If the stack is empty print an error message to the screen,
           otherwise print the details of the customer on top of the stack
           to the screen.
    if i < 1 or i > 3 print an error message to the screen.
    */
}
```

You may assume the existence of the following methods in the Customer class.

```
public String getName();
public String getTitle();
```

[7 marks]

(c) Explain, with the aid of suitable examples, the difference between method *overloading* and method *overriding*.

[7 marks]

(d) Consider the following two classes :

```
class MyClass{
    PrintWriter screen = new PrintWriter(System.out, true);
    public void print(){
        screen.println("This is a method");
    }
}

class AnotherClass extends MyClass{
    public void print(){
        screen.println("This is another method");
    }
}
```

A client class contains the following code :

```
MyClass a = new MyClass();
AnotherClass b = new AnotherClass();
```

What would be printed by the following code :

```
b.print();
```

[2 marks]

```
a.print();
```

[2 marks]

Add code to the **AnotherClass** method which would result in both strings being printed with just the call :

```
b.print();
```

[4 marks]

Question 2.

(a) Explain where a method is visible if it is declared as

- public
- private
- package
- protected

[4 marks]

(b) Explain why a constructor cannot be declared as

- **final**
- **abstract**

[2 marks each]

(c) Consider the following **Abstract Data Type** specification **Queue** for an unbounded queue of integers.

```
Queue(); //Constructor for the empty queue
int front(); //The integer at the front of the queue
void push(int i); //Push integer i onto the end of the queue
void remove(); //Remove the first integer from the queue
Boolean empty(); //Is the queue empty?
```

Write a class **Queue** which implements this ADT. You should use a **Vector** to contain the data and will need probably use the Vector methods **add**, **firstElement**, **isEmpty** and **removeElementAt(int i)**, as well as the default constructor. You may assume that after using the method **removeElementAt** the vector is reduced in size by 1. (i.e. all other elements are moved to fill the gap(s)). Illegal operations should print a warning message to the screen and leave the queue unchanged. Full marks will be given for correct code including return values and visibility modifiers.

[9 marks]

(d) Write four **axioms** which hold for the above ADT.

[8 marks]

Question 3.

- (a) Discuss the object technology concept of **polymorphism** and its usefulness in software engineering using example code and diagrams where appropriate.

[4 marks]

- (b) Explain the meaning of **dynamic binding**. Give details of the problems and benefits of dynamic binding and how **casting** is used to overcome these problems.

[4 marks]

- (c) Explain clearly how object technology lends itself to the concept of **design by contract**. Explain how design by contract is of benefit to the software industry's goals of **software correctness**. (Your explanation should include references to O-O languages other than Java).

[8 marks]

- (d) Explain what is meant by an **abstract** class. Your explanation should include descriptions of **deferred** and **implemented** methods together with reasons why methods should be deferred.

[5 marks]

- (e) What is meant by the following modifiers in Java. Your answer should give reasons / examples for their use.

- final
- static

[4 marks]