QUEEN MARY AND WESTFIELD COLLEGE
DCS/115 INTRODUCTION TO PROGRAMMING

Class Test Friday 17th March 1999 09.00-09.55

Please fill in your name here: .....

College registration number here: .....

Email address here: .....Course Code here:.....

All answers to this test should be written on the test sheet, but you may use spare paper for rough working, or the space on the back page if you cannot fit your answers into the space provided. Answer as many of the questions as you can. This is an "open book" test: you are allowed to access your notes and textbook.

Part A

In this part, all questions refer to the example program distributed separately

Question 1

a) Give an example of a method which is a *constructor* by quoting the line numbers on which the method occurs (for example, if you think the method on lines 104-107 is a constructor, write *104-107*):

b) Give an example, by quoting its line numbers, of a method which *overrides* another method:

c) Give an example, by quoting its line numbers, of a method which *overloads* another method:

d) Give an example of an IS-A relationship between two classes:

e) Give an example of a HAS-A relationship between two classes:

Question 2

In this question, assume the following declarations and initialisations have been made:

```
Dragon draco = new Dragon("red"), hydra = new Dragon("green");
Monster centipede = new Monster(20,50,100);
Monster m1;
Being fred = new Being(12,20);
Weapon excalibur = new Weapon("sword",50);
Weapon w1 = new Weapon("sword",20), w2 = new Weapon("discus",30);
Hero h1 = new Hero(50,150,200,"Xena");
Hero h2 = new Hero(70,120,80,"Percy",w1);
Hero h3;
```

In each part write what the code fragment will cause to be printed:

a)
```
System.out.println(draco);
```

b)
```
System.out.println(centipede);
```

c)
```
System.out.println(h2);
```

d)
```
m1=hydra;
System.out.println(m1);
```

e)
```
h1.pickUp(w2);
System.out.println(h1);
```

f)
```
h3=h1;
h3.pickUp(w2);
h1.pickUp(excalibur);
System.out.println(h3);
```

g)
```
System.out.println(h2.morale());
```

h)
```
draco.attack(centipede);
System.out.println(centipede.morale());
```

i)
```
h2.pickUp(excalibur);
h2.attack(centipede);
System.out.println(h2.morale());
```

j)
```
h2 = new Hero(16,20,100,"Bilbo");
draco.attack(h2);
System.out.println(h2.morale());
```

Question 3

Assuming the same declarations as in question 2, for each of the following code fragments, say whether it is valid or invalid Java, where "invalid Java" is code that will cause a compiler error or warning message when it is compiled:

Valid/Invalid

a) `fred.hit(50,5);` .....

b) `centipede.attack(fred);` .....

c) `fred.attack(centipede);` .....

d) `hydra.attack(fred);` .....

e) `fred = draco;` .....

f) `fred = new Hero(80,130,150,"Fred");` .....

g) `fred.pickup(excalibur);` .....

h) `draco = (Dragon) fred;` .....

i)
```
h3 = fred;
h3.pickUp(excalibur);
```
.....

j)
```
if(fred instanceof Hero)
  System.out.println("Fred is a hero");
```
.....

Question 4

a) A Wyvern is a creature which is just like a Dragon, except it has only two legs. When a `Wyvern` object whose colour field stores `"purple"` is printed it should print "`purple wyvern`". Write a Java class `Wyvern`:

b) Suppose there is an array storing a reference to all the `Being`s in the game:

```
Being everything[] = new Being[MAXBEINGS];
```

where `numBeings` is an `int` storing the number of `Being`s.

Write a fragment of Java code that will cause every `Hero` to have in its `holds` field its own `Weapon` object with `name` field `"spear"` and `power` field `8`.

Part B

Question 1

Suppose you have the following interface:

```
interface List
{
 public boolean isempty();
 public char head();
 public List tail();
 public List cons(char ch);
}
```

defining `List` as an abstract data structure representing lists of characters, with the methods having their standard meanings with lists (i.e `head` gives the first element of a list, `tail` gives the list consisting of all but the first element, and `cons` gives a list whose head is its input character and whose tail is the input list). Suppose you also have an implementation of `List`, `MyList`, where `MyList.empty()` gives a new empty list using this implementation.

Write a method `insert`, which takes a list and a character, and gives the list with the character inserted between the elements of the input list. So if the input is the character `'x'` and the representation of the list `[a,b,c,d,e]`, the output is the representation of the list `[a,x,b,x,c,x,d,x,e]`.

Question 2

Assume the following definition of `LinkedList`

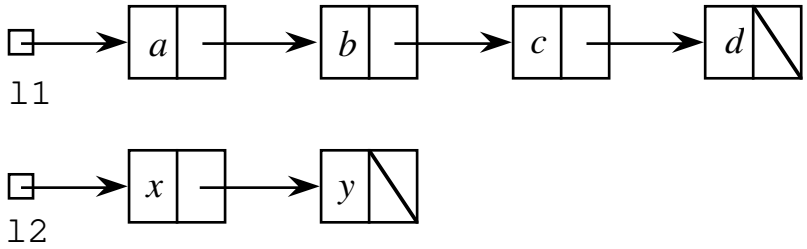
```
class LinkedList
{
 char head;
 LinkedList tail;

 LinkedList(char h,LinkedList t)
 {
  head=h;
  tail=t;
 }
}
```

and a declaration of three variables of type `LinkedList`

```
LinkedList l1, l2, ptr;
```

with the following being a "cells and pointers" diagram showing the position after some code has been executed:



In each of the cases below, assume the situation starts with the above, and sketch a cells and pointers diagram showing the situation after the execution of the code fragment:

a) `l1.tail=l2;`

b)
```
l1.tail.head='z';
l1.tail.tail=null;
```

c)
```
for(ptr=l1; ptr.tail!=null; ptr=ptr.tail);
ptr.tail=l2.tail;
```

d)
```
ptr=l1.tail;
l1=l2;
l2=ptr.tail;
```

e)
```
for(ptr=l1; ptr!=null; ptr=ptr.tail)
   ptr.head='m';
l2 = new LinkedList('k',l1);
```

Please use the space here if you need any additional space to answer any of the questions. For anything written here, please indicate which question it applies to.