QUEEN MARY AND WESTFIELD COLLEGE
DCS/115 INTRODUCTION TO PROGRAMMING

Class Test Friday 6th December 1999 09.35-10.55 (or 16.05-17.25)

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.

1) In each of the cases below, write what would be printed when the fragment of Java code is executed (assume we have the declaration `int a, b, c`):

a)
```
a=1;
b=3;
c=4;
System.out.println(a+b*c);
```

.....

b)
```
a=5;
b=a*3;
a=b-6;
System.out.println(a--);
```

.....

c)
```
a=3;
b=10;
c=b%a;
System.out.println(c+" "+a);
```

.....

d)
```
a=2;
b=5;
b*=a+1;
System.out.println(b+"+"+a);
```

.....

2) Suppose that in a class called `Read` we have a static method called `number` which takes no arguments, and returns the next integer typed into the window, or throws a `NumberFormatException` if what is typed is not an integer.

Thus `x=Read.number()` reads a number and puts it in variable `x` (or throws a `NumberFormatException`).

To the right, write a piece of code which uses this method to read a series of numbers, ending when something which is not an integer is typed in. When it ends, it prints the sum of the numbers typed in.

3) The class `DogBot` to the left below models a simple artificial dog. Suppose we have two variables `rover` and `patch` of type `DogBot`. For each code fragment on the right, write what it will print:

```
class DogBot
{
 int hungry,tired;

 public DogBot()
 {
  tired=1;
  hungry=3;
 }

 public void eat()
 {
  hungry-=2;
 }

 public void sleep()
 {
  tired-=2;
  hungry++;
 }

 public void play()
 {
  tired+=2;
  hungry++;
 }

 public String noise()
 {
  if(hungry<4&&tired<4)
     return "woof ";
  else if(tired>hungry)
     return "snore ";
  else
     return "whine ";
 }
}
```

a)
```
rover = new DogBot();
rover.play();
System.out.print(rover.noise());
rover.play();
System.out.print(rover.noise());
rover.eat();
System.out.println(rover.noise());
```

---

b)
```
rover = new DogBot();
rover.eat();
rover.play();
System.out.print(rover.noise());
rover.play();
System.out.println(rover.noise());
```

---

c)
```
rover = new DogBot();
patch = new DogBot();
rover.play();
patch.eat();
System.out.print(rover.noise());
System.out.println(patch.noise());
```

---

d)
```
rover = new DogBot();
rover.play();
patch = rover;
patch.eat();
rover.play();
System.out.print(rover.noise());
System.out.println(patch.noise());
```

---

4) Assuming the class `DogBot` of question 3,

a) Write a method `hungrier` to be added to the class `DogBot` that compares two `DogBot` objects, and returns a boolean value indicating if one has a higher value in its `hungry` field than the other.

b) Make use of the method written above to write a code fragment which causes the `eat` method to be called repeatedly on whichever of the `DogBot`s referenced by `patch` and `rover` is the hungrier, until the other one is the hungrier.

5) With each of the following fragments of code, give possible values of the integer variables `a`, `b` and `c` which would cause the indicated word to be printed:

a)
```
if(a<b)
   if(b>c)
      System.out.println("Apple\n");
   else
      System.out.println("Orange\n");
else
   System.out.println("Plum\n");
```

Word	a	b	c
Apple	..	..	..
Orange	..	..	..
Plum	..	..	..

b)
```
if(a==c&&b>c)
   System.out.println("Apple\n");
else if(a<c)
   System.out.println("Orange\n");
else if(b!=c)
   System.out.println("Plum\n");
else
   System.out.println("Cherry\n");
```

Word	a	b	c
Apple	..	..	..
Orange	..	..	..
Plum	..	..	..
Cherry	..	..	..

c)
```
if(a>c++&&b>c)
   System.out.println("Apple\n");
else if(a==c)
   System.out.println("Orange\n");
else
   System.out.println("Plum\n");
```

Word	a	b	c
Apple	..	..	..
Orange	..	..	..
Plum	..	..	..

6) In the space to the right, write a method which will print a pattern like that below. The method should have a parameter which gives the size of the pattern to be printed (expressed as the number of stars in the middle line, 7 in the example)

Use the space below if you need any additional space to answer any of the questions.