Have a good break!!!

Question 0. (answer omitted)

Question 1.

```
/** Change this rhino's father to dad.
    Precondition: dad, if not null, is a male.*/
public void setFather(Rhino dad){
     if (father != null) {
        father.noc= father.noc - 1;
     }
     father= dad;
     if (father != null) {
       father.noc= father.noc + 1;
     }
}
```

Question 2.

```
/** Remove all female fathers from r's ancestral tree.
    Note: r may be null, in which case do nothing. */
public static void remove(Rhino r) {
   if (r == null)   return;

   // r != null, so it may have parents

   remove(r.getMother());

   Rhino f= r.getFather();
   remove(f);

   if (f != null && f.getGender() == 'F') {
     r.setFather(null);
   }
}

/** Remove female fathers from rhinos in v[k..]. */
public static void remove(Vector<Rhino> v, int k) {
   if (k == v.size())
       return;

   // v[k..] has at least one element
   Rhino r= v.get(k);
   remove(r);
   remove(v, k+1);
}
```

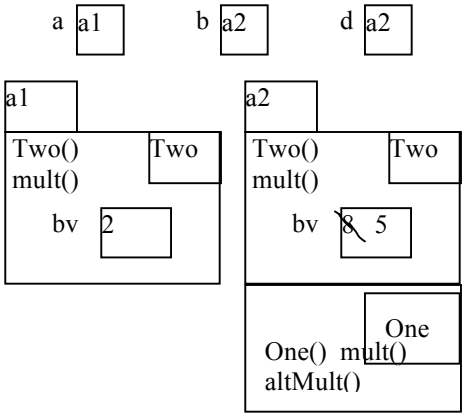
Question 3.

```
/** Remove female father's from rhino trees
    in v[k..]. */
public static void remove(Vector v, int k) {
   if (k == v.size()) {
      return;
   }

   // v[k..] has at least one element
   Object ro= v.get(k);
   if (ro instanceof Rhino) {
      Rhino r= (Rhino)ro;
      remove(r);
  }
  remove(v, k+1);
}
```

```
/** = "r is a Rhino and r's birthdate is the
       same as this Rhino's birthdate" */
public boolean equals(Object r) {
   if (!(r instanceof Rhino))
      return false;
   Rhino r1= (Rhino) r;
   return mob == r1.mob  &&  yob == r1.yob;
}
```

Question 4.



4c: 1: 9; **2:** 10; **3: false**; **4: true**; **5: true**; **6:** 10.

Question 5. (a) Make a class abstract so that objects of the class cannot be created. Make it abstract by sticking keyword **abstract** after **public**.
(b) Make a method abstract so that any subclass must override it. Make it abstract by putting **abstract** after the access modifier and using ";" for the method body.

(c) "double d= 1;" is legal. The cast of 1 to double format takes time.

"int i= 5.000;" is illegal.

"Object ob= **new** JFrame();" is legal. The newly created JFrame object has to be cast to class Object. This takes no time; it is only a "matter of perception", so that the object is viewed as an Object instead of a JFrame.

"JFrame jf= ob;" is illegal. Object ob has to be cast from Object to JFrame and such downward casts have to be done explicitly.

"Animal a= (Animal) ob;" is legal; it will compile. However the cast will cause the program to abort with an Exception because ob is not an instance of Animal.