University of Toronto
CSC148/A58f—Introduction to Computer Science, Fall 2000

Midterm Test: Solutions

Duration: 50 minutes

Aids allowed: None

Make sure that your examination booklet has 6 pages (including this one). Write your answers in the spaces provided. Write legibly.

Family Name: ____________________

First Name: ____________________

Student # : ____________________

Tutor (circle one):

Alexandra Geoff Daniel

1.	________	/ 17
2.	________	/ 14
3.	________	/ 10
Total	________	/ 41

Question 1. Short answers/Multiple-Choice
[17 marks]

Circle the most appropriate answer in the following questions. Parts (a)-(g) are 2 marks each. Part (h) is 3 marks and requires a written answer.

(a) Every class implements one abstract data type.

TRUE $\boxed{\text{FALSE}}$

Ans.: A class does not have to implement an ADT; an ADR can be implemented by more than one class.

(b) There are four modifiers for visibility of classes in packages: `public`, `private`, `protected` and default.

TRUE $\boxed{\text{FALSE}}$

Ans.: Only `public` and default are used for classes.

(c) You need to import a package to access any of its `final` classes.

TRUE $\boxed{\text{FALSE}}$

Ans.: Importing a class is independent of whether it is final.

(d) If we do not use packages, `private` and `protected` have the same meaning for visibility of methods.

TRUE $\boxed{\text{FALSE}}$

Ans.: `public`, `protected` and default are same, `private` is different.

(e) Static methods are more efficient than instance methods.

TRUE $\boxed{\text{FALSE}}$

Ans.: No difference in terms of code space or execution speed.

(f) Every class has at least one constructor.

$\boxed{\text{TRUE}}$ FALSE

Ans.: The default constructor (inherited from Object if not explicitly there)

(g) If a class defines two methods with the same name in a class, this is called

POLYMORPHISM $\boxed{\text{OVERLOADING}}$ CASTING SHADOWING OVERRIDING

(h) An abstract class does not need to implement bodies for any of its methods. Give the major reason why we need interfaces at all when we could just use abstract classes instead.

Ans.: Since a class can only extend from one parent class, whereas a class can implement multiple interfaces, interfaces frees us from being tied to a class hierarchy.

Question 2

[14 marks]

Consider the following class:

```
public class ListNode {
    Object data;
    ListNode link;

    public ListNode(Object data, ListNode link) { // constructor
        this.data = data;
        this.link = link;
    }
}
```

(a) [2 marks] Suppose we want to create a linked list called `myList` with elements ⟨*"apple"*, *"banana"*, *"mango"*, *"orange"*, *"pear"*⟩, where the node containing "apple" is at the front of the list, and the node containing "pear" is at the back. Write Java code that will create this list. Assume "apple", ..., "pear" are strings.

Ans.:

```
ListNode myList = new ListNode(new String("pear"), null);
myList = new ListNode(new String("mango"), myList);
myList = new ListNode(new String("orange"), myList);
myList = new ListNode(new String("banana"), myList);
myList = new ListNode(new String("apple"), myList);
```

(b) [3 marks] Now consider the following method.

```
  public static ListNode whatAmI (ListNode first, int k) {
    ListNode oldFirst = first;
    ListNode prev = null;
    ListNode next = first;
    for (int i=0; i<k; i++) {
       prev = next;
       next = next.link;
    }
    first = next;
    prev.link = null;
    while (next.link != null) {
       next = next.link;
    }
    next.link = oldFirst;
    return first;
 }
```

Assuming that a call to `printList(first)` prints the contents of the list referred to by `first` in the order as they appear in the list, and that `myList` is as created in part (a), what will the following line of code print?

```
printList( whatAmI(myList,3) );
```

Ans: `orange pear apple banana mango`

(c) [4 marks] What does the method `whatAmI` do?

Ans: `whatAmI` moves the first k objects of the list referred to by first and moves them to the end of the list. Thus, it is doing "rotate by k".

(d) [3 marks] Write an appropriate precondition for this method.

Ans: $0 < k <$ the number of objects in the list referred to by first.

(e) [2 marks] Currently `whatAmI()` is a static method that receives `first` as a parameter. Instead consider defining a class `LinkedList` which keeps track of the front of the list as an instance variable. If we want to rewrite `whatAmI()` as an instance method of the `LinkedList` class, we need to make a number of changes. List two changes that will need to be made.

Change 1: We do not need to pass `first`.

Change 2: We do not need to return `first`.

Question 3

[10 marks]

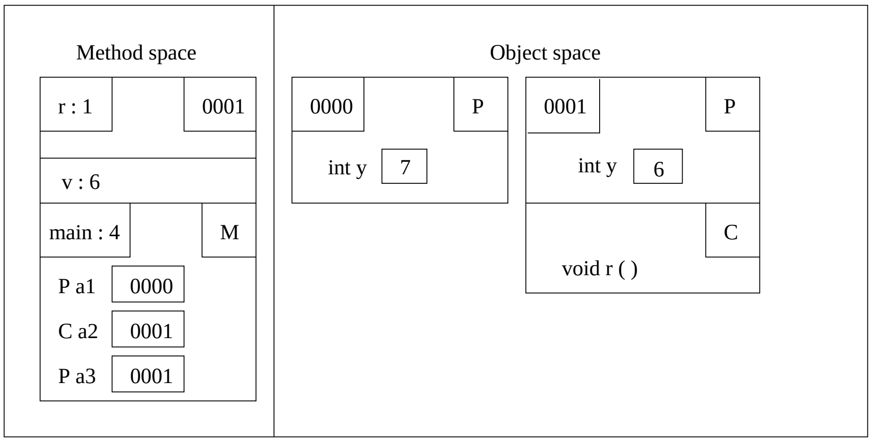
(a) For the classes P and C given below, draw the static space of the Java memory model when these classes are loaded. Put your diagram directly to the right of the given code.

```
public class P {
    static int x = 3;
    int y = 7;

    public static void q(int w) {x = w; }
}

public class C extends P {
    static int u = 5;

    public void r(int v) {
       y = v;
       u = 8;
    }
}
```

Ans:

```
+------------------+
| P |     | Object |
|---+     +--------|
| int x Box{3}     |
| void q()         |
+------------------+

+------------------+
| C |          | P |
|---+          +---|
| int u Box{5}     |
+------------------+
]
```

(b) Write the first 4 lines of the `main` method from the `M` class below so that when you trace your code, you would get the following diagram for the method and object space parts of the Java memory model.




```
class M { // Write the first 4 lines here.
    public static void main(String[] args) {

Answer:
      P a1 = new P();
      C a2 = new C();
      P a3 = a2;
      a1.r(6);
}
```

END OF TEST