CS101 Final Exam, Spring 2008

May 1, 2008

Instructions: This exam is open book. You can use any reference books in the exam. But no computers are allowed. Partial credits will be given in the marking. All the programs MUST be written using JAVA. Be concise and to the point. If you do not know the answer, give your best guess. The total marks are 100 points. You have 1 hour and 15 mins to complete the exam.

1 [20 points]

Trace the following program and write the output.

```java
int n = 2;
while (n < 6)
{
    for (int k = n; k >= 1; k --)
    {
        if (k % 2 == 1)
            System.out.print("*");
        else
            System.out.print("+");
    }
    System.out.println();
    n ++;
}
```

2 [20 points]

Consider the following public section of a Video class. This class represents movie video in a video rental store.

```java
public Video(String title, String category, String media)
{
    Constructor
    Parameters:
    title -- the title of the video
    category -- the category of the video
    media -- the media type of the video
```
public void setLoaned()
    Marks the video as on loan and increases the number of times the video
    has been borrowed

public void setReturned()
    Marks the book as not on loan

public int getTimesBorrowed()
    Returns: the number of times the video has been borrowed

public String getCategory()
    Returns: the category of the video, such as "action", "comedy", "drama",
    "science fiction", etc.

public String getMedia()
    Returns: the media type of the video, such as "dvd", "vhs" or "hd dvd"

Assume that the Video class has been provided to you (you DO NOT need to write the Video
class). Write a static method that has an array of references to Video as input and returns the
average borrowed times of “action” or “science fiction” “dvd” videos. You may need to use the
function equals method in class String. For example, if String a is the same as String b,
a.equals(b) will return true and otherwise false.

3  [20 points]

Write a recursive function to reverse the order of elements in a 1D int array. For example, if the
input array has elements 1 2 3 4, your function will reverse the order of the array so that the array
becomes 4 3 2 1.

4  [20 points]

Write a method to return the data element (an integer) in the second last node in a linked list.
The linked list class has been partially implemented as follows:

    public class List
    {
        class Node
        {
            int data;
            Node next;
            public Node(int data)
            {
                this.data = data;
                next = null;
            }
        }
    }
Node head;
public List() { head = null; }
public void append(int data) {...}

Your function has the interface

public int getSecondLast()

You DO NOT have to write the class. Just complete the method so that it fits into the class List.

5  [20 points]

Consider the following code segment, determine the output of the program to screen when the program is running.

```java
public class Base
{
    protected int data;

    public Base()
    {
        data = 10;
    }

    public void doThis()
    {
        data++;
    }

    public void doThat()
    {
        data--;
    }

    public int getData()
    {
        return data;
    }
}

public class DerivedA extends Base
{
    public DerivedA()
    {
        data = 20;
    }
```
public void doThis()
{
    data += 2;
}

public void doThat()
{
    data -= 4;
}

public class DerivedB extends Base
{
    public DerivedB()
    {
        data = 30;
    }

    public void doThat()
    {
        data -= 5;
    }
}

public class InheritancePlay
{
    public static void main(String[] args)
    {
        final int NUM_OBJECTS = 3;
        int index;
        Base[] hierarchicalObjects = new Base[ NUM_OBJECTS ];
        hierarchicalObjects[0] = new Base();
        hierarchicalObjects[1] = new DerivedA();
        hierarchicalObjects[2] = new DerivedB();
        for( index = 0; index < NUM_OBJECTS; index++ )
        {
            hierarchicalObjects[index].doThis();
            hierarchicalObjects[index].doThat();
        }

        for( index=0; index < NUM_OBJECTS; index++ )
        {
            System.out.println( hierarchicalObjects[index].getData() );
        }
    }
}