University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2007

Practical Test 1 – Test One

Time: 45 minutes

Write a recursive method to determine whether a String is a palindrome or not.

A palindrome is a string that, if reversed, produces the same string. “abba” and “mom” are palindromes while “apple” is not a palindrome.

Hint: Check the extremities and recurse over the middle of the String.

The following methods of the String class may be useful:

- `length ()` to find the length of a String.
- `charAt ( int position )` to extract a single character from a String by position.
- `substring ( int startposition, int endposition )` to extract a part of a String.

You may use the following skeleton program.

```
import java.util.Scanner;

class Test1_Calc
{
   public boolean palindrome ( String s )
   {
      // write your code here
   }
}

class Test1
{
   public static void main ( String [] args )
   {
      Scanner sc = new Scanner (System.in);
      Test1_Calc calc = new Test1_Calc ();
      String s = sc.next();
      if (calc.palindrome (s))
         System.out.println (s + " is a palindome!");
      else
         System.out.println (s + " is NOT a palindome!");
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the Java source files to Vula, using a Zip file if necessary. Name your file **PTest1One.java** or **PTest1One.zip.**

Marking Guide:

- *Correctness: Reduction :20%, Recursion: 20%, Termination: 20%*
- *Comments (Documentation): 40%*

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2007

Practical Test 1 – Test Two

Time: 45 minutes

Write a recursive method to count the number of occurrences of a given character within a given String. *Hint:* Check a small part of the String and recurse over the rest.

The following methods of the String class may be useful:

- `length ()` to find the length of a String.
- `charAt ( int position )` to extract a single character from a String by position.
- `substring ( int startposition )` to extract a part of a String.

You may use the following skeleton program.

```
class Test2_Calc
{
   public int count ( char c, String s )
   {
      // insert your code here
   }
}

class Test2
{
   public static void main ( String [] args )
   {
      Scanner sc = new Scanner (System.in);
      Test2_Calc calc = new Test2_Calc ();
      String s = sc.next();
      char c = 'X';
      System.out.println ("Number of Xs is " + calc.count(c, s));
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the Java source files to Vula, using a Zip file if necessary. Name your file **PTest1Two.java** or **PTest1Two.zip.**

Marking Guide:

- *Correctness: Reduction :20%, Recursion: 20%, Termination: 20%*
- *Comments (Documentation): 40%*

University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2007

Practical Test 1 – Test Three

Time: 45 minutes

Write a recursive method to reverse a String.

Hint: Deconstruct a small part, reverse the rest and reconstruct in reverse order.

The following methods of the String class may be useful:

- `length ()` to find the length of a String.
- `charAt ( int position )` to extract a single character from a String by position.
- `substring ( int startposition )` to extract a part of a String.

You may use the following skeleton program.

```
class Test3_Calc
{
   public String reverse ( String s )
   {
      // write your code here
   }
}

class Test3
{
   public static void main ( String [] args )
   {
      Scanner sc = new Scanner (System.in);
      Test3_Calc calc = new Test3_Calc ();
      String s = sc.next();
      System.out.println (s + " reversed is " + calc.reverse (s));
   }
}
```

You may consult your paper notes and textbook, but no electronic resources. You may NOT use a search engine or consult any Web resources (including Vula) or files on your flash disk, hard drive, etc.

Submit the Java source files to Vula, using a Zip file if necessary. Name your file **PTest1Three.java** or **PTest1Three.zip.**

Marking Guide:

- *Correctness: Reduction :20%, Recursion: 20%, Termination: 20%*
- *Comments (Documentation): 40%*