1.Try to print number from 1 to 10 using three different threads. Thread 1 should print 1, 4, 7 and thread 2 should print 2, 5, 8 and likewise.  
  
Code:  
  
import java.util.concurrent.atomic.AtomicInteger;  
  
public class ThreeThreadst0t1t2 {  
  
    AtomicInteger sharedOutput = new AtomicInteger(0);  
  
    public static void main(String args[]) {  
  
  
  
        ThreeThreadst0t1t2 t = new ThreeThreadst0t1t2();  
  
  
  
        ThreadTasks t1 = t.new ThreadTasks(0);  
        ThreadTasks t2 = t.new ThreadTasks(1);  
        ThreadTasks t3 = t.new ThreadTasks(2);  
  
        Thread ts1 = new Thread(t1);  
        Thread ts2 = new Thread(t2);  
        Thread ts3 = new Thread(t3);  
        ts1.start();  
        ts2.start();  
        ts3.start();  
  
    }  
  
    private class ThreadTasks implements Runnable {  
  
        private final int threadPosition;  
  
  
        public ThreadTasks(int threadPosition) {  
            super();  
  
            this.threadPosition = threadPosition;  
        }  
  
        @Override  
        public void run() {  
  
            while (sharedOutput.get() < 9) {  
  
                if (sharedOutput.get() % 3 == this.threadPosition) {  
                    int value = sharedOutput.get() + 1;  
                    System.out.println("Printing output for Thread: "  
                                            + this.threadPosition + "  "  
                                            + value);  
                    sharedOutput.incrementAndGet();  
                   /\* System.out.println("Printing output for Thread: "  
                            + this.threadPosition + "  "  
                            + sharedOutput.incrementAndGet());\*/  
                }  
            }  
  
        }  
    }  
  
}  
  
  
2.  Program to reverse an integer without using string.  
Code:  
import java.util.Scanner;  
  
/\*\*  
 \*   
 \*/  
  
/\*\*  
 \* @author Puru  
 \*  
 \*/  
public class ReverseNumber {  
    public static void main(String args[])  
       {  
          int n, reverse = 0;  
       
          System.out.println("Enter the number to reverse");  
          Scanner in = new Scanner(System.in);  
          n = in.nextInt();  
       
          while( n != 0 )  
          {  
              reverse = reverse \* 10;  
              reverse = reverse + n%10;  
              n = n/10;  
          }  
       
          System.out.println("Reverse of entered number is "+reverse);  
       }  
  
}  
  
3.Program to convert a string into a string of opposite charecters. Eg. Cat into cAT.  
Code:  
  
  
public class StringOppositeCase {  
      
    public static void main(String[] args) {  
          
        String input = "Cat";  
          
        System.out.println("Given Input ::: " + input);  
          
        StringBuilder strBuilder = new StringBuilder(input);  
          
        for (int i = 0; i < strBuilder.length(); i++) {  
              
            char c = strBuilder.charAt(i);  
              
            if (Character.isUpperCase(c)) {  
                  
                strBuilder.setCharAt(i, Character.toLowerCase(c));  
            } else {  
                strBuilder.setCharAt(i, Character.toUpperCase(c));  
            }  
   
        }  
        System.out.println("Opposite Case::: " + strBuilder);  
   
    }  
  
}  
4. Check if string is palindrome java.

Code:

import java.util.\*;  
class PalindromeString  
{  
  
 public static void main(String args[])  
 {  
   
 // Create a Scanner object

Scanner s=new Scanner(System.in);  
  
 System.out.println("Enter the string");  
  
 // Read the data

String st1=s.nextLine();  
  
 // Create StringBuffer obj for st1  
 StringBuffer sb=new StringBuffer(st1);  
  
 // Reverse the letters  
 sb.reverse();  
  
 // Check & Print if palindrome  
 if(st1.equals(sb.toString()))  
 System.out.println("Palindrome String");  
  
 }  
  
}

5. Printout all the available combinations of any String like ‘abcde’ like ‘a’, ‘b’, ‘ab’, ‘ba’, etc.

Code:

public class Abcde {   
  
public static void main(String[] args) {   
combinations("abcde");   
  
}   
public static void combinations(String input){

for(int i=0;i<input.length();i++){

for(int j=0;j<input.length();j++){

if(i!=j){

System.out.print(input.charAt(i));

System.out.println(input.charAt(j));   
}   
}   
}   
  
}   
}

Or

import java.util.HashSet;

import java.util.Set;

/\*\*

\* Java Program to find all permutations of a String

\* @author puru

\*

\*/

public class StringHelperCombinations {

public static Set<String> permutationFinder(String str) {

Set<String> perm = new HashSet<String>();

//Handling error scenarios

if (str == null) {

return null;

} else if (str.length() == 0) {

perm.add("");

return perm;

}

char initial = str.charAt(0); // first character

String rem = str.substring(1); // Full string without first character

Set<String> words = permutationFinder(rem);

for (String strNew : words) {

for (int i = 0;i<=strNew.length();i++){

perm.add(charInsert(strNew, initial, i));

}

}

return perm;

}

public static String charInsert(String str, char c, int j) {

String begin = str.substring(0, j);

String end = str.substring(j);

return begin + c + end;

}

public static void main(String[] args) {

String s = "AAC";

String s1 = "ABC";

String s2 = "ABCDE";

System.out.println("\nPermutations for " + s + " are: \n" + permutationFinder(s));

System.out.println("\nPermutations for " + s1 + " are: \n" + permutationFinder(s1));

System.out.println("\nPermutations for " + s2 + " are: \n" + permutationFinder(s2));

}

}

Regards,  
Puru