# Part 1

## Programming Challenge 1:

You are to create two small applications for this programming task; one is called Randomizer, the other Prime Randomizer‘s job is to generate a series of positive random integers and send those to Prime via a distributed queue of integers.

Primes job is to receive the integers and calculate whether the integer is a prime or not and return the answer to Randomizer via a distributed queue ( just a java Queue implementation , no need to implement JMS etc ) that contains the original number and a Boolean; which Randomizer will print to system out.

Points

1. Use only the standard java library

2. Both Applications will run on the same server

3. The system should be as fast as possible

4. The results do not have to be returned in the same order as received

5. You don’t have to go overboard tweaking the prime check

Deliverable

1. A project with code – eclipse or any IDE that you are using. However, eclipse is preferable
2. A Readme.txt should contain the following
   1. What is your design and implementation?
   2. Sample Output
   3. Further work – if you have all the time in the world how would you implementpublic class MyObjectFactory{

private Party party;

public Party getObject(String fullyqualifiedPath)

{

Class c = Class.forName(fullyqualifiedPath);

party = (PersonalParty)c.newInstance();

return party;

}

}

## Programming Challenge 2:

1)      Reverse of a string without reverse method.

2)      Please create a class for the below and send it,

We have a table which has 4 columns as id, name, phone and address.

You need to have a method which will return me the data.

Create a method where you can hard code the data and print the details.

The main aim of above example is how you use data structure.

int firstZero(int arr[], int low, int high)

{

if (high >= low)

{

int mid = low + (high - low) / 2;

if ((mid == 0 || arr[mid - 1] == 1) && arr[mid] == 0)

return mid;

if (arr[mid] == 1) // If mid element is not 0

return firstZero(arr, (mid + 1), high);

else

return firstZero(arr, low, (mid - 1));

}

return -1;

}

int countZeroes(int arr[], int n)

{

int first = firstZero(arr, 0, n - 1);

if (first == -1)

return 0;

return (n - first);

}