



IJICSE

Open Access

Journal Approved by UGC

Contents lists available at [www.ijicse.in](http://www.ijicse.in)

International Journal of Innovative Computer Science &amp; Engineering

Volume 4 Issue 4; July-August-2017; Page No. 17-21

## RANDOM QUESTION PAPER GENERATOR - SOCIAL ENTREPRENEURSHIP: PERSPECTIVES ON EDUCATION

Ms.V.Loganayaki

Assistant Professor, Department of Computer Applications,

KG College of Arts and Science, Coimbatore,

[loganayaki.v@kgcas.com](mailto:loganayaki.v@kgcas.com)

Received 10 May 2017; Accepted 04 July. 2017

### ABSTRACT

Contemporary educations have gone iniquitous. Politics have even made its mark in education, early from getting admission for a college and making graduation simple by letting the common assessment exams and semester questions out and pulling the concept of education into the No-Talent state!

It should be fine if there is a random questions generator that would pick questions according to the Designed Application specifically for a College/University with respect to its Blue Print and Weight-age and relying on feasible algorithm that is very easily applicable in any simple code block and in any language – B-Randomization Algorithm (it is introduced in this paper). It is obvious that most of the institutions (every institution at times) are different with its Question paper pattern which varies from Subject to Subject and Department to Department. When this is the case every college can have a uniquely designed application or software that does this work (Randomly generating Question papers). This means that for every college, there has to be a firm which builds this software and does its maintenance and it gets paid in turn and this provides the opportunity to students/youths for their own start up. This idea is highly workable since every subject pattern is not the same as other subject, every subject have individual pattern in every exams like, common assessment test, model exam, semester and etc., a single subject could have varying pattern in different semesters and every college might have a change in its pattern according to which this software is designed. The probability stays high and there has to be individual question paper generator or a single generator could have changeable patterns where we could toggle to the required pattern.

B-Randomization Algorithm is all about making use of easily retrievable values which are un-predictable and implementing the resultant number (for count) in the series for picking up a number in random. Once the number is picked up again this process is done to get another value in the series (series will have numbers/elements excluding previously picked numbers/elements). This way the series or numbers or elements are finally randomized.

This method of randomization provide ease for the developers to design the software or application in any platform implementing the randomization codes on their own instead calling or using a complex packages/methods. This process of generating question papers ensures universities confidentiality and faculties doesn't have an option for going unfair by letting the questions out as those questions wouldn't be known to them anyway before. Taking the education system into a righteous path and making the Indian youths a better entrepreneur, starting from their career from inception (probably every college/university would need an entrepreneur to accomplish this. May be, every department in a college could require such entrepreneur at times to design their own Random Question Paper Generator).

**Keywords:** Contemporary, Iniquitous, Assessment, Unique, Randomization, Inception.

### Introduction and Problem Statement:

As Education system has gone wicked by the corrupted work ethics of Staffs there is a strict need for an overcome through some method which could no way indulge teachers/faculties

into such activities. Staff members take up doing a crime by letting question papers out to their close students which is more of a fraudulent act. Parents or Guardians get their child to study in a School or College or University to explore their

curriculum's king size. When a child fits to be a pet kid for a staff then the child experiences ease in all the way getting excuses and preparing for exams and tests with relieve as the staff would give away the question papers. In certain point of time when the student had to face some other staff the situation becomes critical and also that kid would have become blunt minded expecting spoon feeding all time. More pathetic is that even Talented kids lose their chance or stand when such practices happen which is much more to be concerned about.

To overcome such a big issue of Question paper frauds a Random Question Paper Generator is introduced which gets all the possible questions in all the possible categories (like 1 Marks, 2 Marks, 3 Marks, 5 Marks, 10 Marks, 20 Marks, etc.,). When the generator is activated random question from the series is generated where this process eliminates the manual Question Paper preparation which leads to question paper leakage and other malpractices.

This Random Question Paper Generator works on the basis of B-Randomization Algorithm which makes randomization feasible in any programming language and in any platform without any constrains and without calling or making use of any pre-defined methods of randomization. This Algorithm provides easiness for the programmer while designing the Random Question Paper Generator Software/Application.

### Social Impact and Solution Summary:

This Random Question Paper Generator generates questions in random from the given series of questions through B-Randomization Algorithm.

### B-Randomization Algorithm:

Step 1 : Declare a Variable and assign the address of the variable to another variable say, addvar and delete the variable after assigning the address.

Eg : int a;

addvar = &a; (addvar = 12ef1)

Step 2 : The Characters (only characters and not numbers) in the variable addvar are converted into 0s (using looping statements).

Eg : addvar = 12001

Step 3 : Binary equivalence of the characters in addvar is found. Total number of 1's are counted and the count is added to addvar. Step 3 is Optional and can be implemented in consideration to the complexity to accomplish the conversion to binary with no difficulty.

Eg : addvar = 12001

When addvar = 12ef1. Binary equivalence for e = 01100101 (totally 4 1s) and f = 01100110 (totally 4 1s)

addvar = 12001 + 4 + 4 => addvar = 12009

Step 4 : Get the current second and add it to addvar.

Eg : addvar = 12009

Current\_Second = 43

addvar = 12009 + 43 => addvar = 12052

Step 5 : Find the Sum of all the digits in variable addvar that gives 1 final result.

Eg : addvar = 12052

result = 1 + 2 + 0 + 5 + 2 => 10 => 1 + 0

result = 1

Step 6 : According to the size of the total elements (2 digits, 3 digits, etc.,) this process is executed from Step 1 to Step 5 (2 times for 2 digits obtaining 2 results. First result is the first digit and the second result is the second digit. Same is for 3 and more digits.).

Eg : result = 1

The first digit is 1

By next time, for instance 4 might be the result where, 4 will be the second digit.

Final\_Result = 14

Step 7 : Select the element (Position : Final\_Result obtained from Step 6. The position is counted from the first element in the list/series and it continues like a chain till the Final\_Result value is reached.) from the List, Display it, Delete the element and move the post elements one position prior occupying the deleted element space.

Step 8 : If Total Number of Elements = -1 then the procedure stops processing. Else Repeat from Step 1.

This B-Randomization Algorithm is named considering all the other randomization algorithms to be complex and categorising them all into A. Where this is easily understandable randomization algorithm which uses only basic concepts to find a random value (address retrieving and current seconds retrieving) and can be easily coded in any language. This makes this algorithm to stand unique from other randomization algorithms and to categorise it into a separate group – B and hence named to be B-Randomization Algorithm.

This algorithm gives a feel of comfort for any programmers who plan to build Random Question Paper Generator Software/Application. This Random Question Paper Generator takes the edification society to an enhanced level avoiding education pull down. This can be done as

complete software consisting of different templates for each question paper pattern where set of all possible questions have to be entered as inputs according to its mark and chapter segregation. These inputs are considered and a random question from the set is outputted. Random question is selected according to the result that is obtained through B-Randomization Algorithm. When the count or number of required questions is given the Random Question Paper Generator itself generates those required number of questions. Once a question is selected and outputted it is then deleted from the list avoiding repeated question frequencies. Check cases (conditional checks) can be given for avoiding repeated questions that have come already in previous year question papers. This Random Question Paper Generator can be even in a form of desktop applications (number of apps), each can have only one single template which could generate only one question paper. This means every department will be in need of few apps with which they can generate question paper in their required pattern inputting into the correct app.

There might be an occurrence of 2-3 questions from same topic. If that is the case to be rectified then that particular questions can be manually eliminated or any such cases can be overcome just by using conditional checks.

This Random Question Paper Generator acts to be just the concept or proposal and it can even be applied by means of web-portal. A university can have all such templates or patterns designed in a portal where department heads can login to the university portal, paste the set of questions in the required template which then gives the complete question paper. For a software or application development C# or .net knowledge is suffice and it would be the best preference than any other language or technology. PHP and MySQL knowledge would be suffice for a web-portal development. This shows that achieving this is pretty easier since the languages which are mentioned as prerequisites are trouble-free to explore and it is effort and involvement that matters.

#### **Random Question Paper Generator – a Social entrepreneurship model and accessing the social impact:**

Being a workable concept/idea this would be handled by maximum of the institutions for which that institution's students itself can be an

entrepreneur developing and maintaining the software. It is witnessed that thousands of entrepreneurs will be fashioned having the count of institutions (schools, colleges, universities and other educational institutions) in mind.

In further view after the implementation of this idea education becomes wise standard and talented students come front flourishing in their academics and the chance of teacher's mindset reflecting in the question paper (intended to prepare it easily or hard) gets to the brink. This idea takes Indian education into a different dimension and in a diverse perspective.

#### **Entrepreneurship value proposition:**

There exist randomization algorithms like Monte Carlo randomization algorithm and Las Vegas randomization algorithm and work becomes over head when there has to be an analysis or research on these algorithms which include verifying polynomial identities, randomized quick sorts, Brute force method, randomized complexity classes, hashing, sampling, symmetry breaking, probability analysis, measurability, Boole's inequality, Markov's inequality, Jensen's inequality, chernoff bound, Azuma's inequality, Method of bound difference, Randomized equality tests and etc. These such algorithms involve complex mathematical calculations and hence are considered into one category A. B-Randomization algorithm stays to be the simplest ever algorithm with no calculation other than addition of address and second retrieval and binary equivalence of the address and hence it is uniquely categorised into category B. This B-Randomization algorithm is very easily usable code block in any programming language. Being a simpler method of randomization it can be used in any simple programs with straightforwardness and easiness.

Example: address of variable x is 20f1 (addvar = 20f1) // Address of a variable cannot be predictable  
delete(x) // once the variable is deleted for every instance a new address is predicted  
addvar = 2001 // character is converted into 0  
current\_second = 2 // every language will have an inbuilt method/function for retrieving the second  
BinaryEquivalence\_addvar = 1011100111 // finding the binary equivalence is optional and can be implied if the language facilitate the conversion with easiness

Counting\_BinaryEquivalence\_addvar = 7 // Total number of 1's in the binary equivalence. It can be counted through looping statements.

```
addvar = addvar + current_second + Counting_BinaryEquivalence_addvar
```

addvar is the random number generated

B-Randomization method makes the designer or developer of Random Question Paper Generator to code with less complexity as the core part (randomization) is instantly known. This feasibility in all perspectives makes the developer to go ahead with the process of creating or building the software or application with interest.

The need for this concept of Random Question Paper Generator to avoid internal faults and frauds is being stretched and this awareness is wide spread among the minds of university/college delegates. It is the auspicious time for youths to introduce the software or application to the institutions which would be highly welcomed by them and that is the success of this research paper enabling youths to rise becoming a flourishing entrepreneur.

#### Target Clients:

Institutions (Schools, University, College, Examination Board, Departments in Colleges and other educational institutions) are the exclusive target clients for whom this idea is a sole dedication.

#### Competitors Analysis:

Student's initiative by proposing this idea to their own institution could make them an entrepreneur. The threatening factor will be their hindrance factors that make them still by not going ahead with the proposal to the institution. When this is the case competitors seek chance by proposing the initiative before you do.

#### Financial Sustainability:

Internet acts to be the prime source for any erudition. Any programming novice individual or a person with just basic C and C++ knowledge can easily explore .net framework and a bit of html scripting knowledge will do great in exploring PHP. Through internet surfing it is probable in getting as much as tutorials we need with keen explanation on the language. There are video tutorials available for free to learn the App/Software development. For developing any application SDK (Software Development Kit) is mandatory and even that is available for free. It is the individual's interest, passion, involvement,

knowledge, devotedness and eagerness that is the sole requisite for building such an application or software. As the worst case if an individual is not okay with the ample of available tutorials taking up a course in any institution would not cost more than 13,000 which in turn makes them eligible to earn the same per month after their course completion. More than all the considerations it is the initiative that a student has to take for becoming an aspiring entrepreneur. Regarding to this proposal of Random Question Paper Generator if a student initiates approaching many colleges then he might earn in lakhs, if he does this for a university then the university might have number of subjects with varying pattern and number of affiliated colleges so, this might even give them a good turn over. It is also doable for a student to approach his or her own college where they are pursuing and do this application or to do it for his/her own department which would give them a good earnings. Maintenance is also a factor for which the college/university will have to pay the student in each course of time and royalty may be an inclusion for earning prospects. This is more of an investment less initiative which involves shrewdness and programming skills that could make a man earn tons of bucks.

#### Least Case Estimation:

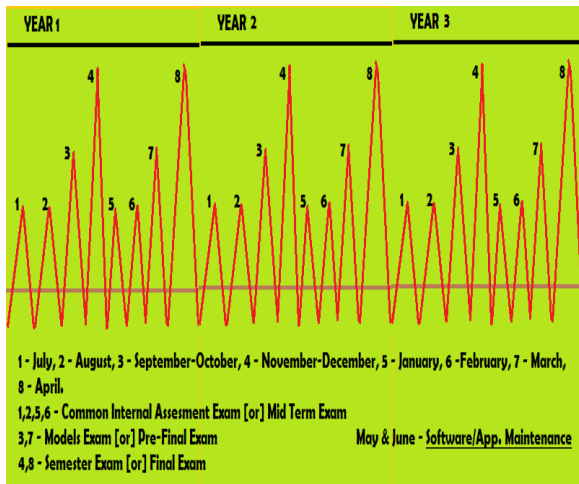
Every college has a minimum of 7-8 Departments. For each semester there will be at least 4-8 subjects for each class. Every department will have 3-4 year students. For an internal exam :  $8*8*4 = 256$  approx. Question papers is required. Same count is for model exam and semester exam. Approx. Amount paid by college to faculties for setting question paper :

For 1 internal exam : Rs. 15

For 1 model exam : Rs. 25

For 1 Semester exam : Rs. 100 approx.

When this is the situation institutions would never mind even paying a bit more for such Random Question Paper Generator and would encourage such question papers. For a student it is **one time work** to develop the software/application and life time earning with it. It is also demandable for bucks in the name of maintenance.



**Fig.1: Cash Flow Diagram for 3 years**

For all the years the same graph prevails denoting a good growth until or unless there is a hike by the internal institution.

**Risk Assessment:**

There might be replications of questions from previous year question papers or more questions might be from the same topic which then is a lug down for the question paper quality. These risks

are easily handled by conditional checks which eliminate those questions and replaces with some other question which is not of that type. It can even be eliminated my man just by removing or erasing the particular question and to replace it with another question which B-Randomization Algorithm generates. Testing for the application is highly recommendable to avoid bugs or crashes. If it is carried out in form of an application then it has to be done in a plat form which is not open source which ensures the security and quality of the app.

**References:**

1. [theory.stanford.edu/~pragh/amstalk.pdf](http://theory.stanford.edu/~pragh/amstalk.pdf)
2. [en.wikibooks.org/wiki/Algorithms/Randomization](http://en.wikibooks.org/wiki/Algorithms/Randomization)
3. [www.ece.northwestern.edu/~nickle/randAlg/Karp91.pdf](http://www.ece.northwestern.edu/~nickle/randAlg/Karp91.pdf)
4. [www.cs.yale.edu/homes/aspnes/classes/469/notes.pdf](http://www.cs.yale.edu/homes/aspnes/classes/469/notes.pdf)