

Parametric Questions in E-Assessment And E-Teaching

Field of the Invention

[0001] The present invention relates to assessment and teaching/training in an electronic-learning or -training environment. In particular, this invention relates to a system and method for random generation of parametric values in questions for assessment and/or teaching of a student/trainee in a virtual environment.

Background

[0002] Electronic- or e-learning is becoming an important learning platform in educational, training institutes and/or corporations to complement traditional classroom learning. E-learning can provide individual or group learning and flexible lesson time, even across geographically separate regions.

[0003] Conventional learning systems provide training and/or assessment. A training system typically allows students/trainees to practise by repeated learning. However, in an assessment system, a student is typically allowed one or a limited number of attempts. Both types of systems either train or assess students by getting them to complete a set or more of exercises based on multiple-choice, true/false and/or fill-in-the-blanks questions.

[0004] In some institutions, it is typical to provide a large number of questions in a databank, and randomly select questions for a training/assessment session. In this way, the opportunity for students to copy or adapt their answers from their peers is minimized. This would also discourage students from learning from their seniors and defeat the purposes of an assessment. However, the types of questions that can be set may still be limited. For example, for multiple choice or fill-in-the-blank type of questions, the time required to solve such questions, say, for an engineering problem is preferably short so as to allow the students to try many questions. This also means that the type of questions that may be set is limited to the comprehension- and/or knowledge-based types of questions.

[0005] Application-based type of questions requires in-depth analysis. Typically application-based type of questions comprise many sub-questions related to the answers to the preceding sub-questions. This type of question cannot be handled by multiple-choice and/or true/false approaches. Application-type questions are conventionally handled by the fill-in-the-blanks. This fill-in-the-blank application-type approach also means that the sub-questions cannot be randomly arranged with other questions. A limitation of the conventional system is that it can only check for the correctness of the students' inputs by comparing them with predefined answers. Such a system is thus not able to emulate the way a human educator assesses a student.

[0006] Another difficulty in a conventional system is the assessment of individuals in a group assessment. Often, all the members would be given the same score irrespective of the individual's contribution to a group.

[0007] In an assessment, for example, at the end of a teaching/training programme, it may not be fair to allow only one attempt per student/trainee. However, re-sitting for an assessment means that a student could learn from the earlier attempts. The scores obtained by a student after a number of attempts would not be a fair evaluation compared with those students who are assessed based on one or fewer numbers of attempts. Further, a student being disadvantaged in a way because of some circumstances may require separate assessment criteria.

[0008] A further disadvantage of conventional assessment is that students/trainees typically sit through a two- or three-hours period at a sitting. It is advantageous if an assessment session can be staggered into separate sessions at the convenience of the students/trainees who then must complete it within a time allocation.

[0009] It can thus be seen that there exists a need for a system and method for improving on electronic assessments/training that can minimize if not overcome the disadvantages of the conventional systems.

Summary

[0010] In one embodiment of the present invention, a system for electronic training and assessment is provided. This system comprises: a processing unit which includes a database for storing a plurality of questions from a plurality of topics of assessment/training, with each question being designed with a variable parameter; and a user interface between a user and the processing unit; wherein the processing unit generates a random number to provide a unique variable parameter each time a user selects a topic of assessment/training through the user interface.

[0011] In one embodiment of generating a parameter, the random number is generated based on the characters of a user identification number. In another embodiment, the random number is based on the date and time information when a user logs-in to the system, selects a question from the plurality of topics of assessment/training and/or re-attempts the question. In yet another embodiment, a parameter is selected from an array of standard numbers based on the nearest value or a sequential count from a predetermined position in the array.

[0012] In another embodiment of the electronic training and assessment system, it further comprises a timer, an attempt counter and an e-guide. The system updates each user performance in a user account.

[0013] In yet another embodiment of the present invention, a method for electronic assessment and training is provided. The method comprises the steps of: storing a plurality of questions from a plurality of topics of assessment/training in an electronic database system, with each question being designed with a variable parameter; generating a random number operable according to a mathematical formula to provide a unique parameter; randomly issuing a user with a question from said plurality of questions in a topic of assessment/training selected by the user from said plurality of assessment/training topics; generating an answer to each question with the unique parameter provided to the user and comparing the answer supplied by a user with said

system generated answer; and awarding a score for each correct answer and updating the user score in a user account.

[0014] In an embodiment for generating a random number, it involves extracting each character of a user identification number. In another embodiment, it involves extracting the date and time information when a user logs-in to the system, selects a question from said plurality of assessment/training topics and/or re-attempts the question. The extracted information is operated according a mathematical formula to provide a unique parameter.

[0015] In one embodiment embodying an application-type or a non-application-type question, when the user's score is below a predetermined score set in the database and it is the user's second and subsequent attempt, the system launches an e-guide to assist the user during each attempt. However, the user score is progressively reduced for each additional attempt.

[0016] In an embodiment embodying an application-type question, which comprises a plurality of inter-related sub-questions, when an answer to a sub-question does not tally with the system generated answer, the system according to the above embodiment computes the answer to the next sub-question based on the wrong answer provided by the user to the sub-question, and a score is given for the correct approach in answering said next sub-question.

[0017] In another embodiment of the electronic assessment and/or training method, the user comprises a team. The system according to the above embodiment ensures that a question has not been previously issued to any one of the team members. Each member's individual answer is compared with the system generated answer and a weighted score is awarded to each member for a correct answer.

[0018] In another embodiment where the question involves an application-type question, each individual team member is awarded for the correct approach to a next

sub-question based on the wrong answer to the previous sub-question provided by the user. This is to emulate the human approach to an assessment or training.

[0019] In a further embodiment of the present invention, a question is designed with a set of variable parameters instead of a single variable parameter.

Brief Description of the Drawings

[0020] This invention will be described by way of non-limiting embodiments of the present invention, with reference to the accompanying drawings, in which:

[0021] FIG. 1 shows a system for e-assessment and e-teaching in accordance with an embodiment of the present invention.

[0022] FIG.2 is a flowchart illustrating a log-in process in accordance with an embodiment of the present invention;

[0023] FIG.3 is a flowchart showing the process of generating a random number in accordance with another embodiment of the present invention;

[0024] FIG. 4 is a flowchart showing the process of assessing a user based on a score and a number of attempts for a comprehension and/or knowledge-type question according to another embodiment of the present invention.

[0025] FIG. 5 is a flowchart showing the process of assessing a user based on an application-type question according to another embodiment of the present invention.

[0026] FIG. 6 is a flowchart showing the process of monitoring the time reserve of a user according to another embodiment of the present invention.

[0027] FIG. 7 is a flowchart showing the process of assessing each member of a team in completing a team assessment according to another embodiment of the present invention.

Detailed Description

[0028] An electronic based learning system 10, as shown in FIG. 1, typically provides a plurality of computer terminals or stations 16 linked to a processing unit 12. The processing unit 12 houses the learning system 10. After a student/trainee has signed up for a course, the student/trainee is provided with an account, user identification number 20 and, preferably, a password. Each time a student wishes to start a lesson, one has to log-in to the system 10.

[0029] FIG. 2 shows a flowchart when a login process is started. At step 100, a student selects to START a lesson through a system interface at a computer terminal or station 16. The system 10 prompts a login request at step 104. After a user identification number 20 is entered at step 104, preferably with a password, the login information is sent to the processing unit 12 for authentication. If the system does not identify the user, the system prompts another login request. If the user identification is positively identified, the user identification number 20 is captured and a data-log of the user account is updated. The system then requests the user to select the level of difficulty of his course in step 114 and the choice of learning/assessment topic or paper in step 118 before the course progresses further.

[0030] In another embodiment of the log-in process at step 118, the user is prompted to select whether to start a learning module or an assessment module. In another embodiment, the selection of a learning or assessment module is predetermined by an administrator or instructor. This selection will become clearer in the later part of this description.

[0031] The system 10 has a database or databank 14 for storing the teaching and/or assessment materials. The database or databank 14 forms a part of the processing unit 12. These teaching and/or assessment materials include sets of questions 30 at various levels of difficulties, such as Basic, Intermediate or Advanced level. These sets of questions 30 are designed to teach a student, for example, to impart certain knowledge or skill, and/or to assess a student's level of competence. Due to a limit to the number of questions or sets of questions that can be created, a way to overcome this limitation is to add a parametric variable P in each question or question set. In another embodiment, each question or sub-question in a set of questions is created with a set of parametric variables P_a , P_b , etc. For example, in an electrical oscillator circuit problem, the set of variables may be the resistor, capacitor and inductor values. This parameter P , P_a , P_b , etc is varied each time a question or set of questions is issued in a learning or assessment module. An advantage of providing a variable parameter in a question is to minimize the opportunity for students/trainees from comparing, copying or adapting answers from one another; instead, it encourages the students to prepare for their e-learning or e-assessment.

[0032] In one embodiment of varying the parameter P , P_a , P_b , etc, the system 10 identifies each character of the user identification number 20 and generates a random number R from the user identification number. For example, a random number R_1 is obtained by operating numerals 1-9 in the user identification number 20 according to a mathematical formula in one embodiment. In another embodiment, the alphabets or zeros in the user identification number 20 are each replaced by a value randomly selected from a predetermined range by the system 10 and another random number R_2 is generated. The second embodiment ensures that random number R_2 and, therefore, parameter P , P_a , P_b , etc, is always different every time the same student is issued a same question. An example of a detailed algorithm for generating random number R is given by Algorithm 1. Algorithm 1 is provided in the Appendix so as not to obscure the clarity of this description.

[0033] In another embodiment for generating a random number R , the system 10 captures the system's date and/or time. The date and time information is then operated

on, for example according to a formula, to provide a random number R. The date and time may be the instant a user has successfully logged-in to the system 10. Alternatively, the date and time may be the instant when a question is issued to a student/trainee as he progresses through a course or module of learning or assessment. Varying the random number R according to the date and time information ensures that the random number R, and therefore parameter P, Pa, Pb, etc, is always different each time a particular student is given the same question. Algorithm 2 is an embodiment for generating random number R based on the system's date and time, and is also provided in the Appendix.

[0034] In an embodiment for varying the parameter P, the system 10 invokes an algorithm to generate a parameter P or a series of parameters Pa, Pb, etc., based on the random numbers generated in the preceding embodiments. This ensures that the parameter P, Pa, Pb, etc. is/are different each and every time a particular student is given the same question. The algorithm for generating a parameter P may be based on a mathematical expression.

[0035] FIG. 3 shows a flowchart for generating a random number R. In step 204, the system 10 extracts the characters from a student's user identification number 20. Each character in the identification number 20 is identified digit by digit. In step 208, each digit is checked whether it is a ZERO or an alphabet. As ZEROs and alphabets are not suitable for mathematical operation, these characters are replaced with a random number selected by the system 10 from a predetermined range. If the condition in step 208 is "NO" or the operation in step 212 is completed, the system 10 extracts the system's date and time information in step 216. In step 224, a random number R is generated based on the results of the operations in step 212, in step 216 or a combination of the results of these operations. For example, these operations may be based on a mathematical expression or formula. In step 228, a parameter P, Pa, Pb, etc. is generated based on the random number(s) generated in the preceding steps. Generating a random number R according to this invention ensures that the random number R is unique and therefore the parameter in each question is different each time

a student is issued a question by the system 10 and/or each time a question is re-attempted.

[0036] In generating the parameter P, Pa, Pb, etc. additional rules may be assigned in each question or set of questions and a suitable type of parametric value is used in the question. For example, the parametric value for use in a question may be a decimal number, a whole number, or a number within a range. For instance, a resistor value in an electrical engineering problem must be selected from a standard array of whole numbers. Algorithm 3, in the Appendix, is an embodiment for rounding off a random parametric value for a resistor. Algorithm 4, also in the Appendix, is an embodiment for finding a nearest resistor value to parameter P, Pa, Pb ... from a standard array of resistor values. In another embodiment, a resistor value may be selected based on a count sequence, for example as counted from the P position from, say, the first value in an array.

[0037] In an implementation of the present invention, the process flow between reference points A and B in FIG. 3 can be interspersed between the corresponding points in the flowchart shown in FIG. 2. In another embodiment, the process flow between reference points A and B in FIG. 3 can be executed after the process in step 118 in FIG. 2 is completed.

[0038] FIG. 4 shows a flowchart for a part of a learning cycle as a student/trainee continues, for example from step 118 in FIG. 2. In step 220, the system 10 issues a question or question set 30 with variable parameter P, Pa, Pb, etc..

[0039] In step 232, the student attempts the question and submits an answer to the system 10. The system 10 generates its answer with the variable parameter P, Pa, Pb,... that was/were issued to the student. The system 10 then compares the student's answer with the system's answer and a score 40 is given for a correct answer. In step 236, the score is compared with a pre-defined score 44. If the student's score 40 is above the pre-defined score 44, the system proceeds to step 240 in which the student's score is updated in the user account. In step 240, a penalty is applied to the student's score if it

is the student's second and subsequent attempt. In one embodiment of the penalty system, the student's score is reduced by a factor X if it is the student's second attempt and the predefined score is progressively reduced by the factor X for each additional attempt at a question 30. If the student's score 40 in step 236 is below the predefined score 44, the system checks whether this is the student's first attempt in step 244.

[0040] If the decision in step 244 is "NO", an attempt counter 48 is incremented in step 248. The system 10 then identifies that the student is weak and an electronic- (or e)-guide 52 is launched to assist the student in step 248. Alternatively or in addition to the attempt counter 48, an e-guide counter 56 is incremented. After performing step 248, the system 10 jumps back to reference point D.

[0041] In a second (and subsequent) attempt, the system 10 allows the student to re-attempt the question 30 with the (old) variable parameter P, Pa, Pb.... In another embodiment, the system 10 executes steps 216 and 224, and generates new variable parameter P, Pa, Pb,... in step 228. As the student progresses through the second (and subsequent) attempt, the system checks on the student's score against the (accumulative) predefined score in step 236, and the system 10 proceeds to update the predefined score in step 240 or re-launch the e-guide in step 248 depending on the performance of this student.

[0042] Going back to the step 244 in the student's first attempt, when the decision in step 244 is "YES", the system 10 allows the student another attempt at the same question 30 in step 252. In one embodiment of FIG. 4, the system 10 allows the student to attempt the same question 30 with the old parameter value P, Pa, Pb,... by continuing with step 232 after executing step 252. In another embodiment of FIG. 4, steps 216, 224 and 228 are interspersed between steps 220 and 232 and the second (and subsequent) attempt includes capturing the current date and time information in step 216. With the latter embodiment, new random number R and parameter P, Pa, Pb, are generated in the same question 30 each time a student re-attempts a question 30. In an alternative embodiment of FIG. 4, the attempt counter 48 is incremented in step 252 instead of in step 248.

[0043] In an implementation of the embodiment in FIG. 4, the process flow from reference point C in FIG. 4 may begin from the corresponding point in FIG. 2.

[0044] The process flow in FIG. 4 may end in an endless loop if a student does not provide a correct answer after many attempts. To avoid such a scenario, the system prompts a user to terminate a course session once the attempt counter 48 indicates that a predetermined number of attempts have been made.

[0045] The process flow for a teaching module has been illustrated in FIGs. 2 and 4 and explained in the preceding paragraphs. The same process flow for an assessment module can be implemented in another way by controlling the number of attempts a student is allowed to try. The number of attempt can be controlled by the attempt counter 48 reaching a pre-set number. Re-attempts may also be allowed only for a student/trainee performing below a pre-defined grade or score. Alternatively or in addition to the attempt counter 48, a timer 50 may be provided. With these embodiments, a student's performance can be more fairly assessed based on the number of attempts and/or time period allocated. Another advantage of this invention is that in the assessment module, all the students are provided with questions of identical level of difficulty albeit with different variable parameter(s) P, Pa, Pb, In this way, assessment is fairer as compared to getting students to answer different questions randomly selected from a question databank.

[0046] The process flow illustrated in FIG. 4 is easy to illustrate for a simple question or question set 30 in a teaching or assessment module. Such a question 30 may be multiple-choice, true/false, or fill-in-the-blank types. In another embodiment of the present invention, an approach to application-type questions 30A is illustrated in FIG. 5. This application-type approach emulates the way a human educator evaluates a student in a learning or assessment session. This application-type questions 30A have many inter-related parts, and the answer to one part is related to another part. This poses a problem to the approach in e-learning and e-assessment.

[0047] As shown in FIG. 5, the process flow of the application-question approach starts from step 1220. In step 1220, the system 10 issues an application-type question 30A with a number of parameters P1, P2, etc. corresponding to the number of inter-related parts are generated.

[0048] In step 1232, the student completes the answers to the question 30A and submits them, for example, by clicking on a submit button on the system interface at a station 16. In step 1300, the system 10 computes the answers to all the parts to the question 30A. In step 1305, the system checks the student's answer to part 1 of the question 30A with that of the system's answer. If student's answer is correct, the system proceeds to step 1315. In step 1315, the system 10 updates the score 40 for the 1st part of the question 30A in the student's account and proceeds to check the answer to the second part.

[0049] If the student's answer to part 1 is incorrect, i.e. the decision in step 1305 is "NO", the system 10 re-computes the answer to part 2 from the student's wrong answer to part 1 in step 1310. In step 1315, the system 10 checks the student's answer given (in step 1232) to part 2 against the system's answer (in step 1300) and also against the system's regenerated answer in step 1310. This ensures that the student is not unduly penalized for a mistake in arriving at the answer to part 2 for a mistake in part 1 of the question 30A. Such a mistake could be a careless mistake or a mistake in entering the answer at the computer terminal/station 16. Advantageously, this embodiment of the present invention allows the student to be awarded a score 40 for the correct approach taken in answering part 2 of the question 30A. In this way, the present invention emulates a human approach in teaching or assessing a student/trainee.

[0050] Similarly, in step 1325, the student's score 40 for the answer to the (N-2)th part of the question 30A is updated in the student's account for a correct answer and the system 10 proceeds to check whether the student's answer to the (N-1)th part is correct. If the student's answer in step 1325 is not correct, i.e the decision in step 1325 is "NO", the system 10 re-computes the answer to part N from the student's wrong answer to part (N-1).

[0051] At the penultimate step of checking the student's answer to part N of the question 30A in step 1335 in FIG. 5, a score 40 is awarded for a correct answer and no score for an incorrect answer. As in the previous step of checking the student's answer (in steps 1305, 1315 and 1325), the system checks the student's answer against both the system answer (in step 1300) and the system's re-generated answer (in step 1330). The system 10 then proceeds to update the score for the answer to the Nth part and to compute the overall score for the entire question 30A in step 1340.

[0052] From step 1340, the processes of completing the learning session continues with the appropriate steps 236, 240, 244, 248, 252 or 256 in FIG. 4. In one embodiment of FIG. 5, the process following step 252 allows the student to re-attempt the question 30A again with the same parameters P1, P2, etc. by jumping to step 1232 from step 252. In another embodiment, the system 10 moves back to reference point D and executes steps 216, 224 and 228. In step 216, the current date and time information is captured and a new random number R is generated in step 224 and a new set of parameters are generated in step 228 in question 30A for the student's second and subsequent attempts.

[0053] The invention embodying FIG. 5 can be modified for an assessment module. As described for the invention embodying FIG. 4, this modification can be implemented by adding an attempt counter 48 and/or a timer 50 for an assessment module. With the embodiments in FIG. 5, the system 10 is able to identify a student with the ability to solve application-type problems but ending up with incorrect answers due to carelessness. This group of students is, nevertheless, rewarded for the correct approach taken and not being unduly penalized for their carelessness.

[0054] The timer 50 in the modified inventions embodied in FIGs. 4 or 5 can be an up-count or a count-down timer. For example, at the beginning of a training session or assessment, an administrator or instructor presets an allocated time for an assignment to be completed. The allocated time can be the same for everyone or different for anyone who has special needs, for example, due to a physical disability. In addition, extension

of time allocation for everyone or specific person can be made as and when necessary. When a student starts a training session or an assessment for the first time, the system 10 loads the time allocated into a student's time bank. This time bank constitutes part of the user account. When a student starts a course or training session, the timer 50 starts a count down and displays the remaining time from the student's time bank. When the student has submitted an answer, the timer 50 stops and updates the remaining time available into the student's time bank. Whenever the student attempts another question in the assignment or assessment, the timer 50 checks whether there is any time reserve in the student's time bank. The student is allowed to continue if the time reserve is not completely depleted to zero.

[0055] Alternatively, an up-count timer may be used instead of a count-down timer. The use of a timer 50 in this invention provides flexibility to the use of this invention. For example, students with physical disability can be given more time for an assignment or assessment, and students who finish an assignment or assessment in a shorter time can be positively rewarded. In addition, a timer 50 allows students the flexibility of completing the assignment partially and continuing the assignment at the next convenient time so long as the time reserve is not completely used up and the deadline for completing the assignment is not over.

[0056] An implementation of the count down timer 50 as described in the preceding paragraphs is illustrated in FIG. 6. In FIG. 6, after a user selects an assessment in step 2000, the system 10 checks both the allocated time for completing this assessment and a student's account in step 2010. The system then starts the timer 50 to count down from the reserve time in the student's account in step 2020.

[0057] After the student submits an answer to a question in step 2030, the system 10 captures the time spent on this assessment session and updates the reserve time left in the student's time account/time bank. In step 2050, the system checks whether the time allocated is fully utilized. If the decision in step 2050 is "YES", the system prompts the student to end his assessment session in step 2080. If the decision in step 2050 is "NO", the system prompts the student whether he wishes to continue. The student has a choice

to end his session or continue with another assessment in step 2070, and the process continues accordingly.

[0058] In another embodiment of the present invention, system 10 provides for the assessment of a group of students/trainees as shown in FIG. 7. In step 3000, each team member logs-in at separate stations 16 and selects the same or different assignments to be assessed depending on an instructor. The system 10 then selects a team question in step 3010 and checks whether the question has been attempted by any one of the team members before. If a question 30A has been attempted by a team member before, the system re-selects another question from the system database or bank 14. If a question has not been attempted by a team member before, a question 30A with a variable parameter P, Pa, Pb.... is issued to each student in step 3020 and each student proceeds to complete the assignment and submits their respective answers. Each student's score and time used up are captured in the student's record and time bank, also in step 3020.

[0059] In one embodiment between reference points D and E in FIG. 7, the system 10 carries out the processes between and inclusive of the steps 216 and 240 as illustrated in FIG. 4. In another embodiment between reference points D and E in FIG. 7, the system 10 carries out the processes between and inclusive of the steps 216 and 240 in FIG. 5. In either embodiment, the system 10 proceeds to step 3030. In step 3030, the system 10 checks whether all the team members have completed their respective assessment or assignment. If not all the students have completed the assignment, the system 10 checks whether any team member has selected a new question in the assignment in step 3040. If a new question is selected, the system 10 then checks whether there is sufficient time reserve in a student's account/time bank in step 3050. If all the time allocated is depleted, no new questions are selected by any one of the students and/or all the members have completed the respective assignment or assessment, the system 10 prompts each of the students to end their respective session in step 3060.

[0060] With this embodiment for team assessment, each team member is assessed based on individual performance. For example, the score of a team member is given a

weightage of X% of an individual's score plus (100-X)% contribution from the overall team members' score. By allowing all the team members to attempt different questions from the same e-assignment or for each member to attempt different e-assessment papers in an allocated time, each team can be assessed on how well prepared they are based on the number of questions/assignments a team can complete and within the time spent as recorded by the reserve time remaining in each member's time bank. This team assessment feature encourages members in a team to help each other prepare themselves before taking an e-assessment. In contrast, it discourages "sleeping" members. This discouragement can be more effective if the weightage of individual score is increased and the contribution from the team's score is decreased.

[0061] In another embodiment of a team assessment, a team time bank is created for each team. In another embodiment, the respective data field in each team's time bank is updated for each student's account/time bank, for example after a team assessment is completed.

[0062] In the above description, embodiments of the present invention for teaching and assessment are separately described. In another embodiment, it is possible that no distinction is made to whether a course is for teaching or assessment. If a course is designed for teaching, an administrator or instructor may set the timer as an up-count instead of a down-count. Alternatively, no time limit is set for a student to complete a learning session. If a course is for an assessment, an administrator or instructor merely sets a time limit for the students to complete an exercise.

[0063] While specific embodiments of the system 10 have been described and illustrated, it is understood that many changes, modifications, variations and combinations thereof could still be made to the present invention without departing from the scope of the present invention.

[0064] APPENDIX:

Algorithm 1: Generation of Random Numbers from User Identification

- Read student's identification from student's login

$$ID = \text{Session}(\text{"StudID"})$$
- Extract individual number/character from student's ID.

$$IDNum(n) = \text{mid}(ID, (n), 1)$$
- Replace any of these IDNum that is either a zero or an alphabet with a randomised number ranging from 1 to 9.

$$\text{If Asc}(IDNum(n)) \geq \text{Asc}(\text{"A"}) \text{ And } \text{Asc}(IDNum(n)) \leq \text{Asc}(\text{"Z"})$$

$$\text{OR } IDNum(n) = 0$$

$$\text{Then } IDNum(n) = 1 + \text{CInt}(\text{Rnd}() * 8)$$

Part of the listing showing this coding.

ID = Session("StudID")

IDNum1 = mid(ID,1,1)

IDNum2 = mid(ID,2,1)

IDNum3 = mid(ID,3,1)

IDNum4 = mid(ID,4,1)

IDNum5 = mid(ID,5,1)

IDNum6 = mid(ID,6,1)

IDNum7 = mid(ID,7,1)

If Asc(IDNum1) >= Asc("A") And Asc(IDNum1) <= Asc("Z") OR IDNum1 = 0 Then
 IDNum1 = 1 + CInt(Rnd() * 8)

If Asc(IDNum2) >= Asc("A") And Asc(IDNum2) <= Asc("Z") OR IDNum2 = 0 Then
 IDNum2 = 1 + CInt(Rnd() * 8)

If Asc(IDNum3) >= Asc("A") And Asc(IDNum3) <= Asc("Z") OR IDNum3 = 0 Then
 IDNum3 = 1 + CInt(Rnd() * 8)

If Asc(IDNum4) >= Asc("A") And Asc(IDNum4) <= Asc("Z") OR IDNum4 = 0 Then
 IDNum4 = 1 + CInt(Rnd() * 8)

If Asc(IDNum5) >= Asc("A") And Asc(IDNum5) <= Asc("Z") OR IDNum5 = 0 Then
 IDNum5 = 1 + CInt(Rnd() * 8)

```

If Asc(IDNum6) >= Asc("A") And Asc(IDNum6) <= Asc("Z") OR IDNum6 = 0 Then
IDNum6 = 1 + CInt(Rnd() * 8)
If Asc(IDNum7) >= Asc("A") And Asc(IDNum7) <= Asc("Z") OR IDNum7 = 0 Then
IDNum7 = 1 + CInt(Rnd() * 8)

```

Algorithm 2: Generation of Random Numbers from System's Date and Time

- Read system's date and time


```

MyMonth = Month (Now)
MyDate = Date (Now)
MyMinTime = Minute(Now)
MySecTime = Second(Now)

```
- Change all these time and date information into a 2 digit number


```

Month, date, minutes and seconds to be set as a 2 digit number
Jan set to 01, Feb set to 02,... Dec 12

If Len(MyMonthTime)<2 Then MyMonthTime="0"&MyMonthTime
If Len(MyDateTime)<2 Then MyDateTime="0"&MyDateTime
If Len(MyMinTime)<2 Then MyMinTime="0"&MyMinTime
If Len(MySecTime)<2 Then MySecTime="0"&MySecTime

```
- Extract information of every digit for month, day, minute and second


```

Month(n) = CInt(Mid(MyMonthTime,n,1))
Date(n) = CInt(Mid(MyDateTime,n,1))
Min(n) = CInt(Mid(MyMinTime,n,1))
Sec(n) = CInt(Mid(MySecTime,n,1))

```
- Replace any of these digits that are zero with a randomised number ranging from 1 to 9.


```

If Month1 = 0 Then Month1 = 1 + CInt(Rnd() * 8)
If Month2 = 0 Then Month2 = 1 + CInt(Rnd() * 8)
If Date1 = 0 Then Date1 = 1 + CInt(Rnd() * 8)
If Date2 = 0 Then Date2 = 1 + CInt(Rnd() * 8)
If Min1 = 0 Then Min1 = 1 + CInt(Rnd() * 8)
If Min2 = 0 Then Min2 = 1 + CInt(Rnd() * 8)
If Sec1 = 0 Then Sec 1 = 1 + CInt(Rnd() * 8)
If Sec 2 = 0 Then Sec 2 = 1 + CInt(Rnd() * 8)

```

Part of the listing showing this coding.

```

MyMinTime = Minute(Now)

If Len(MyMinTime)<2 Then MyMinTime="0"&MyMinTime

Min1 = CInt(Mid(MyMinTime,1,1))
Min2 = CInt(Mid(MyMinTime,2,1))
If Min1 <= 0 Then Min1 = 1 + CInt(Rnd() * 8)
If Min2 <= 0 Then Min2 = 1 + CInt(Rnd() * 8)

MySecTime = Second(Now)

If Len(MySecTime)<2 Then MySecTime="0"&MySecTime
Sec1 = CInt(Mid(MySecTime,1,1))
Sec2 = CInt(Mid(MySecTime,2,1))
If Sec1 <= 0 Then Sec1 = 1 + CInt(Rnd() * 8)
If Sec2 <= 0 Then Sec2 = 1 + CInt(Rnd() * 8)

```

**Algorithm 3: Generation of a Randomised Parametric Variable
(Example that of a resistor value)****Minimum: 1,072,655, Maximum: 9, 653, 895**

```

'-----
' Randomize Resistor value in Ohm
'-----

RandMOhm1 =
Round(
CLng(((5^1*(IDNum1)))+(5^2*(IDNum2)))+(5^3*(IDNum3)))+(5^4*(IDNum4)))+(5^5*(
IDNum5)))+(5^6*(IDNum6)))+(5^7*(IDNum7))+5^8*(Month1+Month2)+5^7*(Date1+
Date2)+5^6*(min1+min2)+5^5(sec1+sec2))

```

**Algorithm 4: Generation of Randomised Standard Values
Example standard value of resistors****Define array**

```

'-----

```

' Array of Standard Resistor Values

```

a Resistor
= array
(10,11,13,15,16,18,20,22,24,27,30,33,36,39,43,47,51,56,62,68,75,82,89,92,100,110,120,
0,
130,150,160,180,200,220,240,270,300,330,360,390,430,470,510,560,620,680,750,820,
890,920,1000,1100,1200,1300,1500,1600,1800,2000,2200,2400,2700,3000,3300,3600,
3900,
4300,4700,5100,5600,6200,6800,7500,8200,8900,9200,10000,11000,12000,13000,15000,
16000,18000,20000,22000,24000,27000,30000,33000,36000,39000,43000,47000,51000,
56000,62000,68000,75000,82000,89000,92000,100000,110000,120000,130000,150000,
160000,180000,200000,220000,240000,270000,300000,330000,360000,390000,430000,
470000,510000,560000,620000,680000,750000,820000,890000,920000,1000000,1100000,
1200000,1300000,1500000,1600000,1800000,2000000,2200000,2400000,2700000,
3000000,3300000,3600000,3900000,4300000,4700000,5100000,5600000,6200000,
6800000,7500000,8200000,8900000,9200000)

```

' Find nearest standard resistor values function

```

Function FindNearestValue(arr, randNum)
    Dim x, result
    Dim curDiff
    result=""

    'iterate through the array
    For x=0 To UBound(arr)

        'calculate current difference:
        curDiff=CLng(arr(x))-CLng(randNum)

        'return value if equal:
        If curDiff=0 Then
            result=arr(x)
            Exit For
        End If
    End For

```

'value is not equal, check for nearest standard resistor value larger than the 'randomised value.

x=0

```
Do While curDiff<0
    x=x+1
    curDiff=Abs(CLng(arr(x))-CLng(randNum))
Loop
    result=arr(x)

FindNearestValue=result

End Function
```

CLAIMS:

1. A system for electronic-assessment and/or -training comprising:
 - a processing unit which includes a database for storing a plurality of questions from a plurality of topics of assessment/training, with each question being designed with a variable parameter; and
 - a user interface between a user and the processing unit;
 - wherein the processing unit generates a random number operable to provide a unique variable parameter each time a user selects a topic of assessment/training through the user interface.
2. A system according to claim 1, wherein the random number is generated based on the user identification number.
3. A system according to claim 1 or 2, wherein the random number is further generated based on the date and time information when a user logs-in to the system, selects a question from the plurality of topics of assessment/training and/or re-attempts the question.
4. A system according to any one of the preceding claims, wherein the random number generated is used to select a number from an array of standard number based on the nearest value or a sequential count from a predetermined position in the array.
5. A system according to any one of the preceding claims, further comprising a timer.
6. A system according to any one of the preceding claims, further comprising an attempt counter.
7. A system according to any one of the preceding claims, further comprising an electronic guide.

8. A system according to any one of the preceding claims, wherein said database further comprising a user account.
9. A system according to claim 8, wherein said user account comprises a data-log of the user's training/assessment sessions, a score for each session, time spent during each session, and a time reserve remaining from an allocated time after each session.
10. A system according to any one of the preceding claims, wherein said variable parameter comprises a set of variable parameters.
11. A method for electronic-assessment and/or -training comprising the steps of:
 - a. storing a plurality of questions from a plurality of topics of assessment/training in an electronic database system, with each question being designed with a variable parameter;
 - b. generating a random number, which is operable according to a mathematical formula to provide a unique parameter;
 - c. randomly issuing a user with a question from said plurality of questions in a topic of assessment/training selected by the user from said plurality of assessment/training topics;
 - d. generating an answer to each question with the unique parameter provided to the user and comparing the answer supplied by a user with said system generated answer; and
 - e. awarding a score for each correct answer and updating the user score in a user account.
12. A method according to claim 11, wherein said step of generating a random number comprises the steps of extracting each character of a user identification number.
13. A method according to claim 11 or 12, wherein said step of generating a random number further comprises the steps of extracting the date and time information

when a user logs-in to the system, selects a question from said plurality of assessment/training topics and/or re-attempts the question.

14. A method according to any one of claims 11-13, wherein the generated random number provides a selection of a number from a standard array of numbers based on the nearest value or a sequential count from a predetermined position in the array.

15. A method according to any one of the claims 11-14, further comprising the step of comparing the user score with a predetermined score.

16. A method according to claim 15, further comprising the step of determining whether the answer provided by the user is a first attempt.

17. A method according to claim 16, wherein the user is allowed another attempt at the question if the user score is below the predetermined score and it is the user's first attempt.

18. A method according to claim 16, wherein an electronic guide is launched if the user score is below the predetermined score and it is the user's second and subsequent attempt.

19. A method according to any one of claims 15-18, further comprising updating the score to the user account if the user score is above the predetermined score.

20. A method according to claim 19, wherein said step of updating includes updating the actual score for a first attempt and a progressively reducing score for each subsequent attempt.

21. A method according to any one of claims 11-20, wherein said question issued to a user comprises a plurality of inter-related sub-questions.

22. A method according to claim 21, wherein an answer to a sub-question does not tally with the system generated answer, further generating an answer to the next sub-question based on the wrong answer provided by the user to the sub-question, and updating a score for correct approach in answering said next sub-question.
23. A method according to any one of claims 11-22, wherein the user is a team.
24. A method according to claim 23, further checking that a question has not previously been issued to any one of the team members.
25. A method according to claim 24, wherein comparing the answer comprises comparing each individual's answer with the system generated answer and awarding a weighted score to each team member for a correct answer.
26. A method according to claim 24, wherein each team member submits each individual's answer to the question comprising a plurality of inter-related sub-questions, checking whether each team member has approached a sub-question correctly according to the method defined in any one of claims 23-25 and awarding the team member accordingly.
27. A method according to any one of claims 11-26, wherein said variable parameter comprises a set of variable parameters.