Pedagogies to Foster Interest In Computer Programming In Secondary Schools

Presented by

JOHN M. KANDIRI

(Bsc.is- Moi, M.Sc. Is –Nairobi, IT LECTURER STRATHMORE UNIVERSITY)

Presented to

International Computer Science and ICT Conference

Topic of interest

ICT in Education

Contact

STRATHMORE UNIVERSITY

FACULTY OF INFORMATION TECHNOLOGY

P.O BOX 59857 – 00200 NAIROBI

TEL: 020 – 606155

Email : jkandiri@strathore.edu or jkandiri@hotmail.com

Abstract : Preece (1998), in his book *Human-Computer Interaction*[1] notes that understanding the psychology of programming is not easy . This might be explained as the reason most students – those pursuing computer science and also those pursuing other courses have phobia towards programming. But proper introduction to computer science and specifically programming is one of the solutions this paper proposes. Teachers should adopt proper pedagogies in the introductory level, and specifically at secondary schools level so as to nurture students' creativity and analytical thinking. Making students appreciate and love computer programming will foster skills in program design and solution creativity hence developing skilled programmers to cope with the growing ICT industry.

Key words: Teaching Pedagogies, programming phobia, computer studies, software engineering, procedural programming, critical and logical thinking

1.0 Introduction

The author has been an Information technology educator for many years. The first years was in teaching computer studies in a secondary school in Nairobi Kenya. In the last five years , the author has taught Software Engineering, algorithms development and Data structures both in diploma level and also to undergraduate students. He has also supervised several Information Systems (IS) projects for final year students in both diploma and undergraduate.

Algorithms development and IS projects, are practical subjects where students are expected to formulate algorithms, which for example forms the basis of programming. In the questions and answer sessions with students, the author observed that students lacked necessary competence in describing simple algorithms which he(author) posed in class. This is despite the fact that some of these students had pursued computer studies in secondary school and took it as one of the examined subject in their final year (form four) . Their poor performance in algorithm class was not therefore correlated with their performance in the Kenya Certificate of secondary Examinations (KCSE) results where they performed quite well (above B) in computer studies subject. They also had programming phobia as expressed by the author's computer programming colleagues as also as noted when they were to implement the algorithms.

The author there was left wondering why this is so yet the computer science syllabus has Data Structures, algorithms and introduction to programming as some of the topics taught. Why then was there no correlation between the good performance in KCSE computer studies exam and the Data Structures and algorithms and programming subjects?

The above problem is a worrying trend because in the last four years, Kenya has heard the mention of computers and computerization than ever before. Phrases like e-governance, ICT

[1] One of the disciplines that Human Computer Interaction (HCI) borrows from is Software Engineering and therefore programming

policy, police computerization project, need to access tax forms online, Kenya Revenue Authority agents need to log their documents online and other such phrases are so familiar. Institutions like Universities [2], parastatals and major supermarket among other major businesses in Kenya have not been left behind. There is integration of Enterprise resource planning software (ERPs) in most of these sectors. Among these are Uchumi supermarkets, Nakumatt, Safaricom and This shows the that organizations in Kenya have appreciated computers and ICT in general as a strategic resource.

Unfortunately, most of this software is imported. This means as much as we are consumers of ICT, we end up paying heavily for a resource that can be developed locally!. There is therefore insufficient supply of competent programmers despite almost all of our local universities offering degrees and diplomas in computer science. The author notes that Secondary schools in Kenya can form the catchments for computer programming skills which are fine tuned in post secondary school level.

1.1 Training Approaches on the ground

Currently, the traditional methods of teaching are adopted. Unfortunately, the traditional "Chalk-and-Talk" teacher centered approach is not effective in developing analytical and logical skills in students. It is the limitation of the chalk-and-talk approach that had made teachers in other disciplines like science and mathematics to come together and think of the best way to make students appreciate those subjects. One such initiative is "Strengthening of Mathematics and Science in Secondary Education"- WESTERN, EASTERN, CENTRAL AND SOUTHERN AFRICA (SMASSE-WECSA) [3]. SMASSE is discussed further in literature review.

Another problem noted is that most teachers teaching computer science in our secondary schools have not undergone formal training. For example, they
are not graduate teachers but computer science or other disciplines with some diplomas in computer. If therefore the adage "a teacher teaches how he was taught "holds, then we see a limitation here in delivery.

The government seems ready to engage in hardware acquisition and neglecting the human resource development.

The above factors, therefore, are some of the factors that become hindrances to removing programming phobia among students. It is even more appalling because by the final year

[2] NYANDIERE ,Clements's paper , The increasing role of ISs in the Management of Higher Educational Institutions (HEIs) in Kenya , presented during *7th Strathmore Annual ICT Conference on* 9th September 2006 list the local universities and the Information Systems they have integrated

[3] SMASSE-WECSA is discussed at the end of the paper

computer studies programming projects, some schools tend to have computer science teachers program for them while others buy the ready made software from programming "hawkers" who get to know the projects for a given year and come up with products

1.2 Scope

The research will investigate the current teaching methodology in computer studies in selected schools within the Nairobi district. Though gender issue need to be looked at, this research will not analyze the performance of boys versus girls, but within the methodology, the researcher aims to collect data from both girls' and boys' secondary schools equally.

1.3 Significance of the Study

The recommendations proposed in this research can be adopted by ministry of education and therefore secondary schools as the approved approach in teaching computer programming in secondary schools so as to foster a love for programming;

The research can also be used as a basis of further research in the methodologies used in secondary schools training in Kenya so as to enhance (improve) learning in schools. This is especially in private schools

In writing this paper, the author makes the following assumptions:

1. All students can become good programmers if an appropriate methodology is used by the programming teachers to introduce the concepts;

2. All students in secondary schools gained conceptual skills by the age of 12;

3. Students have the same retention capability;

4. Both sexes have equal chances of understanding computer programming;

5. The contribution of personality to programming is not discussed here. In the literature, we can agree that personality differences brought out Bill gate and Linus Torvalds from other students in their classes;

2.0 Literature Review

2.1 Software Engineering

Boehm defines software engineering as "the application of science and math by which the capabilities of computer equipment are made useful to man via computer programs, procedures, and associated documentation" Analyzing this definition, we see that the potential of computer equipment, which this is any device that uses software , for example mobile phone, fridge, car etc can only be made useful by development of computer programs. It is due to integration of computers – what we call programs or software that the Information and Communication Technology (ICT) industry is becoming the biggest employer of late.

These programs include the operating systems and other supporting applications. But for this to be realized , the developer must conceptualize the problem , develop algorithm before embarking on coding . But, Programming is not always easy nor is it as soft job as it looks from outside. For people who have do not share the love for it, it may even at times look . This love for programming comes when developers are able to see the fruits of their work and enjoy it. This is love that can be natured early by teachers using effective approaches to introduce programming to students

2.2 Practical Approach to Teaching

In her article, "How do Children Develop? " Susan Jindrich gives an episode where kids in a school are painting at the easel. Each has a jar of blue paint and another of yellow paint. One boy, who like other boys is naughty mixes the contents of his two jars. The color turns green. He exclaims and attracts the attention of other kids. After enquiring how he did it and him explaining his action, the teacher calls other pupils to go on and test what would happen with other mixtures. This was an introductory class to "colors". The experiment helped the kids chart their discover of colors. Jindrich calls this "discovery learning or hands-on learning"

Research has shown that people remember things better when they learn them by doing. This is even true for adults.

Hands on learning is how all of us were exposed to different things we do. Even learning how to walk! This technique therefore applies to both young and adults. The learner is actively involved instead of just sitting and listening. In order to learn best, school children must also be actively involved in hands-on activities in what they are learning.. Some good examples of hands-on activities are: working puzzles, making collages, painting, measuring ingredients for you as you cook, learning to cook , learning to take care of younger siblings

2.3 Computer Programming In Secondary Schools

Piaget's theory indicates that the formal operational thinking abilities develop around the age of twelve (12) . It is at this age that students start moving from concrete thinking to logical/abstract thinking. This means that the students are able to come up with their own creations.

In his study to determine any correlation between mathematics and procedural programming , White (2003) also tried to determine the development of cognitive skills. He notes that mathematics can be used to determine whether a student has developed cognitive skills. If his study is used here, then we can assume that majority of secondary school students in Kenya have well developed cognitive skills as they do well in mathematics at secondary school level

I agree with Mitchell Kapor[4] that the importance of cognitive skills comes out in the design stage of software development. The software designer is the champion of the user experience as he brings together the world of technology and the world of people and human purposes. The designer has to for example look at user requirements and ask himself how the requirements can be met through software and hardware (for human computer interaction) He therefore looks at the two words, the lay mans world and the technology.

Any system developer appreciates the importance of quality design, where the programmer thinks over a problem and comes up with a technical model. If the teacher delivery on design is poor and cannot take students clearly through the problem conceptualization, then at the end of the day, the product will be poor!

In designing computer programs therefore, the selection of the various components and elements of the application must be driven by an appreciation of the overall conditions of use

---

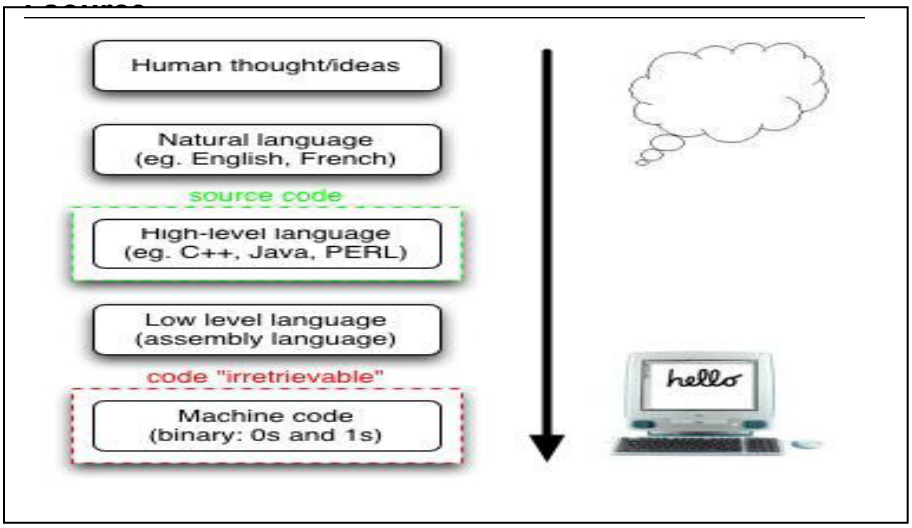
[4] Mitchell Kapor's article "**A Software Design Manifesto" in the book** Bringing Design to Software by Terry Wanograd

and user needs through a process of intelligent and conscious design. How is this to be done? By software designers. The software designer should be the person with overall responsibility for the conception and realization of the program. But there is no direct path between the designer's intention and the outcome. As you work a problem, you are continually in the process of developing a path into it, forming new appreciations and understandings as you make new moves.

Figure 1 below shows the steps one converts human thought into programming. If a student is able to conceptualize the problem clearly enough, by using the natural representation , for example explain the problem in plain English. This will mean converting the problem to more formal method for example structured English becomes easier. For example, if a student is asked how to develop a program to get sum of two numbers, is he able to know that:

i) Some numbers are required for summing?
ii) There is need for the summing action?
iii) There is need for data display?

Figure 1: Steps in the conversion of human thought to machine : (Couros , 2004)



With proper introduction to problem solving, a student is capable of thinking through the process for different programming problems solved..

Ali (2005) clearly states the role of teachers when he says that teachers open the door of knowledge and invite the students in. He continues to note that teachers must create an environment conducive for effective learning and by extent effective transmission of knowledge. He notes that a teaching pedagogy the teacher adopts must look at how to teach,

what to teach and what aids to use in the process. In his article, "Teachers Need Appropriate ICT Skill" Oredo (2006)[5] identifies the following as requirements of a teacher (read quality/ competent teacher) in a given subject:

i) Understand the subject from all dimensions;
ii) Understand the subject's history;
iii) Understand the subject's evolution;
iv) And understand the benefit of the subject to the society.

He continues to point out that the teacher must be conversant with the education system , the psychology of the students and the resources available for efficient delivery of the content. These attributes can be acquired if a teacher undergoes proper training and also is prepared to adapt to changing methodologies of delivery . Unfortunately most of the teachers training computer science in Kenyan schools today are not trained teachers but either pursued computer science or computer related studies and founds themselves in schools. This means that some attributes Oredo spells out might be lacking like understanding student psychology and also best teaching methodologies, if the old adage "teachers teach the way they were taught" suffice, then a limitation can be observed here and therefore do agree with Oredo that teachers must keep abreast with new technologies and new approaches to total delivery.

Fortunately, research has shown that computer training is similar to mathematics. With the developed methodologies of teaching, the teacher can adopt the proposed methods . They can also join the already developed initiatives/forums for training science and mathematics , for example SMASSE.

Teacher training institutions like University of Nairobi, Moi University, Kenyatta University , , Kenya Science Teachers College among others should integrate computer science as a teaching subject. Therefore,

2.4 Strengthening of Mathematics and Science in Secondary Education"- WESTERN, EASTERN, CENTRAL AND SOUTHERN AFRICA (SMASSE-WECSA) and Other initiatives

SMASSE is a program that endeavors to bring together secondary schools teachers who teach mathematics and sciences and discuss how best to deliver the subjects . Its objective, according to the SMASSE website states "The association shall exist for the purpose of Strengthening Mathematics and Science Education at the secondary level, through In-Service Training (INSET), Pre-service training, research, and exchange of information, seminars, joint exercises and all other lawful means to pursue its objectives" It works in collaboration with the Working Group on Mathematics and Science Education (WGMSE). Other such initiatives in the world include Computer Science Teachers Association (CSTA) in America.

[5] Mr. John Otieno Oredo is a computer science teacher at Strathmore School and also a post Graduate student at Strathmore University

Market Wire, an online journal, discusses the initiative by IBM and the Computer Science Teachers Association (CSTA). The initiative aims at supporting and promoting the teaching of computer science and other computing disciplines. This is in line with the CSTA mission of bringing greater exposure for k-12 students to study computer science. One of the beneficiaries of the IBM/CSTA initiative, Shane Torbert , who was a teacher at the Thomas Jefferson High School for Science and Technology in Virginia, participated in a six-week pilot program. After the program, he made the following observation "The structure of the lessons encouraged students to think through the design of a computer program, from problem statement to solution," . He went on to confess that though he previously found the design process generally hard to teach, his instruction on this area improved significantly after the program.

Through the initiative programs, students think of a program design upto final delivery of a system. By the end of the program, teachers, like Shane above , are able to deliver well to students and therefore enhance learning.

2.5 Related Study

Sanwar, A (2005) discusses the Scaffolding as one of the most effective teaching pedagogy for undergraduate computer science. He gives an example of the following approach in implementation of scaffolding in introducing computer programming:

1) Presentation of the Problem: The teacher presents a problem that is designed specifically to take into consideration the use of new material as defined in the syllabus for that specific class session. The examples used in class should be realistic. For example, the teacher can pick a mathematics formula learnt in class, and which all students should be familiar with. This can be getting the area of a triangle. This helps students appreciate the role of programming and make it less abstract. When students discover how they can solve simple real world problems, they go out and look for challenging tasks.

2) Problem Formulation: Students, with teacher acting as facilitator, formulate the problem. This requires the understanding of the question as well as the meaning of the problem's terminology, and the identification of its facts: goal, givens, unknowns, condition and constraints. Problem understanding requires the processing of information. The techniques of verbalization and visualization are used to create an initial understanding of the problem. For example, making a drawing, talking, or answering questions about the problem aid the task of problem understanding. Problem formulation evolves with the transformation of the given problem statement into a precisely formulated model. Developing a precise model of the problem is completed by the elicitation and organization of all relevant information and the elimination of irrelevant information.

3) Solution Planning: Students, with teacher acting as facilitator, plan the solution beginning with the development of an appropriate solution strategy. The students consider various

alternatives to determine the course of action suited to achieve the goal of the problem, subdivide the goal into subgoals, and identify the tasks needed to accomplish each subgoal. The relevant information identified in the previous activity is related to the various subgoals and its role and meaning are defined. This enables the students to begin the process of carrying out the strategy to progress toward meeting each subgoal of the problem and eventually producing the complete solution.

4) Solution Design: Students, with teacher acting as facilitator, organize and refine the components of the solution strategy, and define specifications to be translated into code. There are two levels of design. The first is a high-level design where a framework structure for a solution to the problem is produced, typically in visual or outline form. This involves the organization and sequencing of subgoals, the determination of whether the subgoals require further refinement, the establishment of relationships among the various solution components, and the association between data and subgoals. Subsequently, detailed design transforms subgoals into corresponding algorithmic specifications and the solution logic is readied to be translated into programming language code.

5) Algorithmic Walkthrough: Students, with teacher acting as facilitator, prepare to map the algorithmic solution into programming language syntax by going over each line in the algorithm and pointing out the exact language construct required. Any reference to what appears unclear or not yet used or previously covered is marked to be the subject of next activity. A list of such references is maintained.

6) Presentation of Syntax: The only "teacher dominant" activity. This activity brings the class back to the beginning of the textbook chapter of the subject at hand. This part is driven by two objectives: meet the stated goal for this lesson take the solution to its final stages (write the complete program).

7) Implementation: Students, with teacher acting as facilitator, proceed to fill in the gaps and produce a complete program that will run on the computer. This is done by translating the detailed design into programming language code that will be tested and can be executed to produce the result.

8) Testing Students with teacher acting as facilitator perform some testing by hand in the planning and design phases to verify that a reasonable logic is used in the algorithm. However, after the implementation phase, each module developed is tested with data for that module. Once we are satisfied that each module does what it is supposed to, the whole program is tested as a unit. The implementation and testing tasks are evaluated in terms of solution quality and correctness.

In application of Computer-Aided Scaffolding, Sanwar, Ali (2005) notes that teachers can come up with program templates that can help students understand algorithms. A good example is where a template can describe what is happening within a simple iteration. The teacher runs the template and a mark will be indicating a line and any output. By this method,

the students' interest is raised and they follow the demonstration. This is an example of the skills that Oredo was proposing. This, as Al-Imamy , et al (2005) notes, will reduce the time teachers take to explain programming language syntax but give more time to learning a concept. With this, the students' creativity and analytical thinking are developed[6].

The aids can be created in simple application like power point , which is animated or on web interface. This means, during a theory class, the teacher takes the students through the aids as he explain what is happening in a loop.

3.0 Discussion

Programming should be introduced using building block approach. This is where teachers introduce students using simple tasks which they can identify with. For example, instead of the teacher telling students to write a payroll program (which they cannot identify with as they are not working) , identify a mathematical problem students have solved in class for example, getting the area of a geometric shape. Let the students explain what is expected to get area of a geometric shape , for example rectangle. Then the algorithm, then convert this to a programming problem. Computer programming teachers can then use Sanwar scaffolding steps to guide students through implementation of the problem .

Researchers in learning psychologists states that knowledge is something that resides within one's mind , Ali (2005). This one can also observe when teaching. Teachers will agree that one gets surprised with the creativity and answers students provide during a class discussion. The above methods will bring out the students potential after the teacher help them understand concepts by observation and contribution. When the student observes that he has the idea, he feel content by acknowledging the fact that he can discover. The teacher encourages the student to release his potential. This is unlike the chalk-and-talk method where the student sees the teacher as the paragon of knowledge. Actually, it reduces the student to a zombie such that without the teacher, the student is incapacitated.

But for the teacher to have a paradigm shift from the traditional chalk-and-talk , teacher centered approach to pedagogies that will make the students an active participant in the learning process, teachers must be empowered with relevant skills. Below are some of the methods suggested by the author as paramount to ensuring computer programming teachers are ready for the paradigm shift.

Pre-service training. Teacher training colleges should integrate Computer science as a teaching course so as to produce graduate who can deliver the subject better. Introduction of

[6] Al-Imamy , et al , in their paper notes that teachers take excessive time on programming language syntax , leaving very little time for " developing skills in program design and solution creativity"

computer studies as one of the teaching subjects for graduates pursuing teaching degrees and diplomas will ensure that the teachers undergo psychology of learning and special methods in teaching computer studies. Currently, majority of computer studies teachers are computer science graduates with no background in education.

In-service Training and teacher motivation: As noted before, most of the computer studies teachers in secondary schools are not trained teachers. For those teachers already teaching computer science, they should go for in-service training. This will incorporate teaching methods and student psychology among other essential units. It is only with adequate training that teachers can get the confidence for example to come up with templates as describes under Computer - Aided scaffolding pedagogy above.

Government involvement in in-service training: Government should be involved in the in-service training programs as it rolls out ICT in schools. Currently, the emphasis is on computer hardware which does not make a lot of sense without skilled human resource to extract value from the IT resources. During the in-service sessions, teachers will be taken through best teaching approaches for computer studies and programming as a practical subject. To motivate teachers to take this course/training more, the government through the ministry of education and Teachers' Service Commission (TSC) should award recognizable certificates and/or promote the graduates from the teacher upgrade projects.

SMASSE initiative: Computer studies teachers should also be integrated in the CEMASTEA/SMASSE initiative. In these forums, computer studies teachers will be sharing on the best way to introduce computer programming in secondary schools. This is to help them come together and think on best way to deliver the subject so as to initiate love for it. They can also come together with the industry to form initiative like the one IBM formed with Computer Science Teachers Association (CSTA) in USA where IBM supported teacher training in new skills. These initiatives (graduate teachers in computer studies and also in-service training) will ensure that teachers develop deductive skills. This will help improvement in delivering computer studies and more specifically computer programming.

Programming Clubs in schools: Computer studies teachers should encourage formation of programming clubs in schools. Through the club, students will write programs and challenge each others in programming,. These programs can then be presented during the schools Science Congress. he student is also able to understand algorithms better and especially through concept mapping. This is where students are able to create mental models and be able to represent them algorithmically. Through mental models also, the student clearly can develop and communicate ideas freely. The computer-aided scaffolding also helps the student shape his ideas more clearly.

Pro-active methods of assessing final computer studies programming projects: The Kenya National Examinations council and computer studies teachers should be more thorough in computer studies project assessment and ensure that students' effort is awarded and cheating

is punished. This will make the students get focused and develop and interest to understand programming.

4.0 Conclusion

In computer programming, it is necessary for the student to develop programming skills. This is possible if the student appreciate the role of computer programming in coming up with systems and the influence of such systems in every day life. By using Sanwar, A (2005) scaffolding steps, the Bloom's taxonomy of education objectives are met. These objectives include development of:

i. Affective Skills (values, attitudes, and interests)

ii. Psychomotor skills and

iii. Cognitive Skills

In affective skills, by allowing introducing the programming using simple problems, the students is able to actively participates in the learning process, he can also attach a value as they can relate to the application , for example mathematics thus being able to accommodate them within their own schema; comparing, relating and elaborating on what has been learned and finally students will be in a position to value and believe in computer programs as they exercise more complex problems on their own as they appreciate programming. This in essence now exerts influence on their behavior and attitude to programming.

By allowing students to think and do on themselves, teaching develops psychomotor skills. Students appreciate what they can do on their own or in groups

Finally, on cognitive skills, allowing students to move from simple programs to complex programs ensures that they acquire knowledge, they comprehend why they write programs and how to solve a problem and finally, they are able to think through the problem.

It is good to note that once a student has gained good programming skills and found love for computer programming; he/she can be able to:

i. write software from scratch;

ii. Re-engineer existing software

iii. Work with open-source software

But all these is not possible without teachers in computer programming being encouraged and taught to have a paradigm shift from the traditional teacher centered delivery to an approach where the student is encourages to become an active participant in class.

In conclusion, the author believes that Secondary schools in Kenya forms the best catchments to nurture computer programming skills which are fine tuned in post secondary school level. This will see Kenya joining the 21st century global trend in contributing in integration of ICT in development thus making Kenya not just consumers of information systems but also developers.

Reference

1. **Ali , Sanwar (2005)** Effective Teaching Pedagogies for Undergraduate Computer Science, in the online Journal of Mathematics and Computer Science, accessed online on 20th Oct 2006 http://www.findarticles.com/p/articles/mi_qa3950/is_200510/ai_n15868740
2. **Al-Imamy, Samer , Alizadeh, Javanshir and Mohamed A. Nour,** On the Development of a Programming Teaching Tool: The Effect of Teaching by Templates on the Learning Process in the Journal of Information Technology Education , Volume 5, 2006, Available at http://jite.org/documents/Vol5/v5p271-283Al-Imamy115.pdf
3. **Association for the Development of Education in Africa**, Working Group on Mathematics and Science Education, in their "Working Groups" webpage http://www.adeanet.org/workgroups/en_wgmse.html last accessed 22nd Nov 2006
4. **Bandhs,, Thoghts from India, found in his website ,** Available at http://manand.typepad.com/thoughts_from_india/, last updated 30th June 2006.
5. **Barbacci, M**. et al: Technical Report on Quality Attributes , Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, found i=on http://www.sei.cmu.edu/pub/documents/95.reports/pdf/tr021.95.pdf, last accessed 20th Nov. 2006
6. **Computer Resources Support Improvement Program (CRSIP),** Revision 3.0, March 31, 1999: A Gentle Introduction To Software Engineering accesed online on http://www.stsc.hill.af.mil/resources/tech_docs/GISE.DOC
7. **Computer Science Teachers Association website accessed online** http://csta.acm.org/
8. **Couros, Alec** (2004) The Open Source Movement: Implications for Education : http://www.educationaltechnology.ca/couros/publications/unpublishedpapers/Couros-OpenSource-Comprehensives-June30-04.pdf? Last accessed 20th Nov 2006
9. Faculty Center for Teaching and e-Learning, Bloom's Taxonomy of Educational Objectives Accessed online http://www.fctel.uncc.edu/pedagogy/basicscoursedevelop/Bloom.html, last accessed 2nd Jan. 2007
10. **Greene, William**: Growth in Services Outsourcing to India: Propellant or Drain on the U.S. Economy?by U.S. International Trade Commission January 2006: Accessed online http://hotdocs.usitc.gov/docs/pubs/research_working_papers/EC200601A.pdf on 22nd Nov 2006

11. **Jindrich, Susan** , How do Children Develop? "notes http://www.gdrc.org/kmgmt/learning/child-learn.html
12. **LADIPOT, T, LEVY, D and PAZ, T.** Implementing Constructivist Ideas in a Functional Programming Curriculum for Secondary School Students , accessed online http://www.ccs.neu.edu/home/matthias/FDPE99/const.html
13. **Market Wire**, April, 2006 IBM and the Computer Science Teachers Association Help High School Students Boost Computer Science Skills accessed on http://www.findarticles.com/p/articles/mi_pwwi/is_200604/ai_n16138545 last accessed 15th Nov 2006
14. **Mitnick , Kevin and William , Simon** (2005) , The Art of Intrusion : The Real Stories behind the Exploits of Hackers , Intruders and Deceivers, Wiley Publishing, Indianapolis
15. **Nyandiere, C:** The increasing role of ISs in the Management of Higher Educational Institutions (HEIs) in Kenya, presented during the 7th Strathmore Annual ICT Conference on 9th September 2006, unpublished
16. **Panagariya, Arvind**, Globalization and the Outsourcing of Services: The Impact of Indian Offshoring" Last accessed 8th Nov. 2006 http://www.brookings.edu/es/commentary/journals/tradeforum/2005btf_panagariya.pdf
17. **Oredo, J.O** . Teachers Need Appropriate ICT Skill , in the Standard , Wednesday November 2006
18. **Preece, J** (et al) (1999) Human-Computer Interaction, The Open University, England
19. **Pressman , Roger** (2000) Software Engineering : A Practitioner's Approach, 5th ed. McGrawHill
20. **Winograd, Terry**(1996) Bringing Design into Software, Addison-Wesley, Reading, MA