Creating an Environment Supportive of Diversity in Computing

P. Mullins, D. Whitfield, D. Dailey, B. Gocal
Computer Science Department
Slippery Rock University
Slippery Rock, PA 16057, USA
paul.mullins@sru.edu

Abstract— *This paper describes the results of a PDE Keystone/Frederick Douglass Grant that the authors received to incorporate diversity into computing coursework in the areas of Computer Science, Information Systems, and Information Technology. Existing assignments created by professors from all three disciplines were examined for their inclusion of diversity, in its broadest sense. Where little or no diversity was found in course assignments, alternate assignments were proposed that included some degree of diversity.*

Keywords— *computer science; diversity; inclusive environment*

I. Introduction

The authors received a grant from the Frederick Douglass Institute to investigate the incorporation of diversity in computing curricula without adding curricular components. The authors envision the creation of a sample set of assignments that could be used not only within our department, but by other computing departments to create problems for students that were inclusive, that is, supported diversity by making it a normal part of the course work. This limited approach is not intended to suggest that a more inclusionary environment is not necessary or to be sought after; rather, it was intended to provide a means of showing faculty how some degree of diversity could be included in their courses with a minimum of effort or impact on how the class is taught.

Diversity was taken by the investigators in the broadest sense. That is, we included not only gender, people of color, and various ethnic backgrounds, but also included LGBT, religions, and the physically challenged. In this paper, underrepresented people is intended to include any of these.

In addition, the project provided an opportunity for our program to assess the current status of our program by looking at sample assignments submitted as part of an accreditation effort and to attempt to develop a methodology for the creation or modification of assignments to make them more inclusive.

The idea of diversity in the computing classroom is not new – many papers on the topic have been published. One example of a programming assignment used to help students learn the value of diversity and the analysis of the assignment shows the usefulness of using assignments as a conduit for diversity [1]. Many works have been published discussing techniques that attract and retain diverse populations of computing students where diversity includes minorities [2]. The national center for women and information technology [3] works to increase diversity in IT and computing. Works have been published for using widgets to support disabled learners [4].

A. Department Overview

There are three programs (majors) in the Department of Computer Science at Slippery Rock University: computer science (CS), information systems (IS) and information technology (IT). All three programs are accredited by the Computing Accreditation Commission of ABET (www.abet.org).

The department offers 37 courses that are used as requirements or electives in these three majors. One of these (CPSC 490: Independent Study) is used accommodate special situations that arise with students or scheduling. Two (CPSC 498b: Machine Learning and Robotics and CPSC 498c: Compiler Transformation) have not been offered in a number of years. In addition, there are several service-only courses that do not count toward any of the majors. 34 regularly offered courses were analyzed that count as a required or elective course in one or more of the three majors. Naturally, some of these courses count in more than one major. Table 1 provides the number of department courses used in each major. Of these 34 courses, five are also used as part of the university's liberal studies program and two for a basic university computer competency requirement.

TABLE I. BREAKDOWN OF THE 34 MAJOR COURSES

	Information Technology	**Information Systems**	**Computer Science**
Department courses	20	18	21
Required	12	11	10
Elective	8	7	11

Grant support provided by West Chester University of Pennsylvania's Frederick Douglass Institutes of the Pennsylvania State System of Higher Education.

II. METHODOLOGY

Directly addressing diversity, even on a college campus where open discussion and alternative views are honored, can be problematic. For example, one potential assignment called "The game of Choice", refers implicitly to the question of whether sexual preference is a matter of choice. The problem statement for the course assignment is as follows:

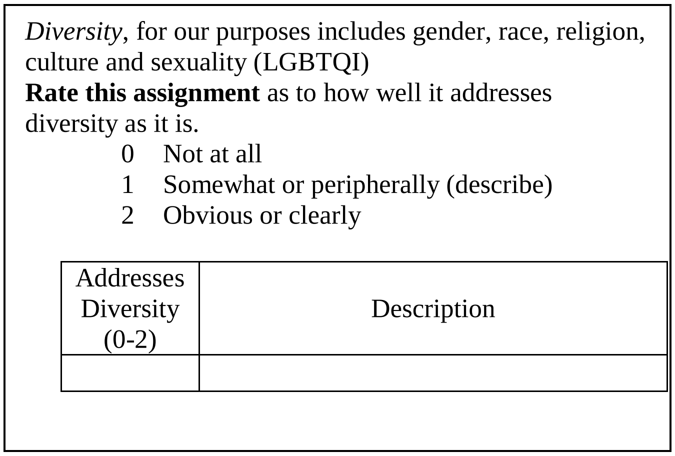
> *"Many still believe that sexual orientation is a matter of choice. It is well known among the ignorant, that proximity to those whose sexual preference varies from one's own may cause one to question one's sexuality or even to change. In Conway's game of life, each cell has one of two states (dead or alive). In ours, each cell will have one of three states: gay, straight or questioning."*

Clearly, people that believe either way could be offended by the problem statement. The problem as originally conceived used people of three religions (Christian, Muslim and Hindu) with somewhat different rules for the state changes. Either formulation of the problem is certain to offend some group. The problem could be "sanitized" for individuals by referring to the states as A, B and C.

Similarly, someone may find that an assignment asking students to create a cash register program for specific items popular in Mexican cuisine as offensive. Similar items may, of course, be chosen from any culture. Once again, the problem is easily "sanitized" by referring to product1, product2, etc. In both cases, the authors view this sanitizing as failure to create an inclusive environment.

As part of the accreditation process, sample assignments, quizzes and exams were collected for 34 of the department courses. Student employees were tasked with completing an electronic form to evaluate the inclusiveness of each of the 135 assignments. The authors then reviewed the students' work. The form consisted of identifying information, definitions, directions, and rubric-like scales. For example, Fig. 1 shows the portion of the form used to determine how well the actual assignment addressed diversity.

In addition, each assignment was rated as to how easily it could be modified to address diversity without changing the pedagogical goals of the assignment, and how easily diversity could be included with a possible modification of the educational goals, without changing them entirely. Next, the authors challenged themselves to generate at least two assignments that included diversity for each course that is used as a requirement or elective in all three majors.



Diversity, for our purposes includes gender, race, religion, culture and sexuality (LGBTQI)
Rate this assignment as to how well it addresses diversity as it is.

0 Not at all
1 Somewhat or peripherally (describe)
2 Obvious or clearly

Addresses Diversity (0-2)	Description

Fig. 1. Portion of data collection form

III. ANALYSIS OF ASSIGNMENTS

A total of 135 assignments were analyzed by the student employees and then reviewed by the authors. It is worth noting that some courses are offered more often than others. Some instructors either did not use or did not submit sample assignments (perhaps using a series of quizzes to evaluate students) and some focus on a small number of projects, while others submitted as many as ten small (lab) assignments. Naturally, some of the assignments are simply worksheets or textbook problems sets, while others are custom-developed projects. Table 2 provides a summary of the number of assignments and the results of the analysis, with some assignments appearing in multiple columns.

That only 24 of 135 (18%) existing assignments addressed diversity as the assignments were submitted, with only four (3%) doing so in obvious or direct manner, may be indicative of a number of things. Some, may even find 18% a surprisingly high number in the rather technical fields of CS, IS and IT. Of the assignments that could be easily changed, 20 are not included in the existing (As Assigned) assignments. We find that having 44 of 135 (33%) of the assignments easily changed to include diversity an encouraging finding. From another perspective, this implies that 33% of assignments could have addressed diversity, while only 18% (3% in an obvious or direct way) actually did.

IV. ASSIGNMENTS WITH DIVERSITY

Investigators challenged themselves to generate at least two assignments that included diversity for each course that is used as a requirement or elective in all three majors. With the exception of one course (CPSC 450: Internship), where only one assignment was devised, we were successful. Most of the assignments (see Appendix A) were uniquely generated,

TABLE 2. ANALYSIS OF EXISTING ASSIGNMENTS

	As Assigned			**Modify w/o affecting goals**	**Modification that might affect goals**
Diversity somewhat or peripherally addressed	20		**Might be changed to address diversity**	18	18
Diversity obviously addressed	4		**Easily changed to address diversity**	27	4

although some were based on existing assignments. That none of the assignments simply referred to an existing assignment may say something about how faculty generate assignments. Presumably, there is a tendency to want new, unique problems – a laudable goal when students often seek "insight" from Internet search engines as a first step to solving a problem.

One of the goals of this study was to create a set of sample assignments that could be used directly, or as the basis for modification, by other faculty at other institutions. As expected, some of the more theoretical classes where a greater challenge and those that included programming somewhat easier. Web programming, in particular, is especially conducive to including diversity, as the assignment often includes or even focuses on images or data which can be selected for diversity. There are four categories of assignments in the courses: problem statements that include examples, assignments that use specific objects, data processing problems, and theoretical problems. Each of these four categories are explained and a sample assignment the emphasizes diversity is provided.

A. *Problems statements that include examples*

Care should be taken to at least maintain neutrality in the use of examples, unless one is seeking to emphasize diversity. The use of he/she or simply using titles, rather than names can help in maintaining neutrality. Selecting Susan or Juan or Juanita as the name of the project manager and John as the programmer implicitly includes an underrepresented group.

A large number of problem statements include one or more examples to help elucidate the problem for the student. Even when the problem does not otherwise address diversity, the examples selected often can. For example, a problem asking the student to sort names (strings) might well be elucidated by the names of famous underrepresented people. Figure 2 provides an example from an IT course.

Web development is inherently more visual than many traditional computing problems and lends itself easily to the inclusion of images. Whether these images are part of the problem, an example, or simply decoration, they allow for an obvious inclusion of diversity.

CpSc 327: Systems Administration and Security
Project 2: Country of Origin

"The highest-piracy countries are Armenia, Bangladesh, Georgia, and Zimbabwe, all over 90 percent." [http://www.telecentre.org/profiles/blogs/the-global-software-piracy] Many attribute the relatively high rates of piracy to social conditions. When it comes to Internet attacks, the US often points the finger of blame at China or other global competitors. Research the issue of where Web-based attacks, phishing URL, and spam originate (or are hosted). Determine the top three countries for all malicious activity. Start with Symantec's Internet Threat Security Reports, but back this up with additional resources.

Fig. 2 Example from Information Technology

CpSc 464: Principles of Concurrent Programming and Operating Systems
Project 1: Threads

You are to write a C program that uses POSIX semaphores and threads to implement a multiple producer, multiple consumer conveyor belt that produces both sopes and molletes. Use the following information to complete the assignment: Two employees, Carlos and Fernanda, consume the Mexican snacks by removing them from the conveyor belt in the order they are produced. The conveyor belt can only have 10 items on the belt at a time with a maximum of 3 sopes permitted on the belt at a time. Additional restrictions: the Mexican snacks are consumed in FIFO order, producers should exit when a total of 100 Mexican snacks are generated, do not use global variables, pass parameters. For proper simulation, you will need to implement parameters that control timing.
-t N the delay in milliseconds between the production of each sope.
-s N the delay in milliseconds between the production of each mollete.
-F N where N is the number of milliseconds that Fernanda needs to place an item in its wrapper
-C N where N is the number of milliseconds that Carlos needs to place an item in its wrapper

Output each time a Mexican snack is produced and each time a Mexican snack is consumed. At the end of the run, print a summary of how many sopes and molletes were produced and how many Mexican snacks Carlos and Fernanda consumed.

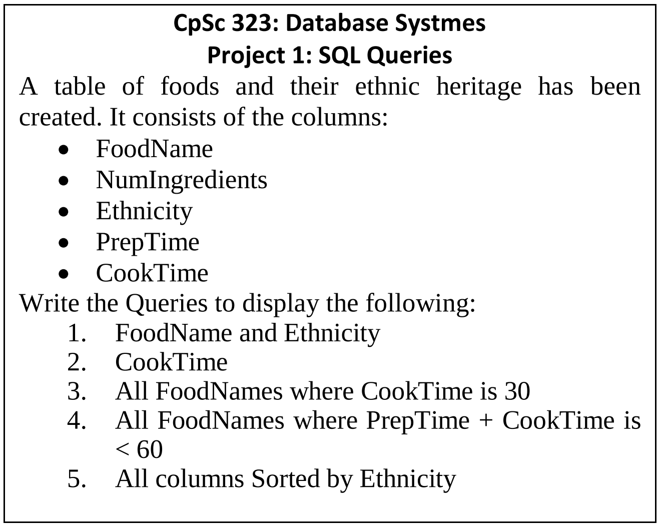
Fig, 3 Example from Computer Science

B. *Problems that use specific items*

In many programs the problem is made more real to the student by identifying specific objects. These might be objects that are being purchased or sold, organized spatially or searched for. In any case, example objects may be selected that in some way represent diversity, items that are generally thought of as being representative of a culture (as shown in Fig. 3). Of course, care should be taken not to select items that represent a stereotype.

C. *Data processing problems*

All computing programs address data in some way, but many are designed to read, process and output data. The sample data set often offers an easy opportunity to include diverse names of people or items as seen in Figure 4.



CpSc 323: Database Systmes
Project 1: SQL Queries

A table of foods and their ethnic heritage has been created. It consists of the columns:

- FoodName
- NumIngredients
- Ethnicity
- PrepTime
- CookTime

Write the Queries to display the following:

1. FoodName and Ethnicity
2. CookTime
3. All FoodNames where CookTime is 30
4. All FoodNames where PrepTime + CookTime is < 60
5. All columns Sorted by Ethnicity

Fig. 4 Example from Information Systems

V. CONCLUSION

For 30 of the 34 courses at least two assignments that easily addressed diversity were created. Programming and discussion courses were easy to develop assignments for. However, some courses do not seem to lend themselves to more inclusionary assignments. In particular, Computer architecture, Computer Organization, Theory of Computation, and Compiler Design assignments did not fit the generic approaches described above. When the assignment is to build an adder, diversity may seem unattainable. However, an assignment to write an assembly language with instructions for non-Western cultures could be developed. Hence, great imagination can be used to coerce an assignment to be more inclusive.

For those courses that assignments cannot be created, it is generally worthwhile to discuss with the students a particular person from some underrepresented group as a means of motivating the assignment. Although this is separate from the actual assignment and does use at least a small amount of class time, it is an excellent way to make students aware of the contributions to the discipline from underrepresented people. Dr. Richard A. Tapia [5] or Maria Klawe[6]could be used for theoretically courses; Admiral Grace Murray Hopper [7] could be used for hardware or compiler courses.

REFERENCES

[1] M. Wick. "Using programming to help students understand the value of diversity". Proceedsing of SIGCSE 2009

[2] J. Cohoon and L. Tychonievich. "Analysis of a CS1 Approach for Attracting Diverse and Inexperienced Students to Computing Majors". Proceedings of SIGCSE 2011. Pgs 165-170

[3] National Center for Women and Information Technology. http://ncwit.org

[4] V. Gkatzidou, E. Pearson, S. Green, and FO. Perrin. "Widgets to support disabled learners: a challenge to participatory inclusive design" OzCHI '11: Proceedings of the 23rd Australian Computer-Human Interaction Conference. November 2011.

[5] Tapia, Richard. http://www.caam.rice.edu/~rat/

[6] M. Klawe, http://www.hmc.edu/about1/administrativeoffices/officeofthepresident1/bio.html

[7] Hopper, Grace. http://www.history.navy.mil/bios/hopper_grace.htm

APPENDIX A: SAMPLE ASSIGNMENTS

Each department course used in one of the majors is listed here along with a sample assignment for that course. Courses are identified by title and include an indication of which major (CS, IS or IT) uses the course and whether the course is required (R) or elective (E). A full listing of the curriculum for each major is available at http://cs.sru.edu. The complete list of example assignments is available at http://cs.sru.edu/~mullins/diversity/AppendixA.pdf.

CpSc 130: Introduction to Programming and Information Systems (IT-R, IS-R)
Project 2: Best Movie Page

Create a web page that lists various categories of films, including at least action, foreign, romance, comedy, and gay/lesbian. Using the images provided as icons (select others as needed), allow the user to choose a category. The link takes the user to a separate category page for each genre you have chosen. In each category, add at least two well-known films. If you don't personally know of a film that should be listed (your opinion), do a key word search at imdb.com. Each Movie listing must include a title and a short description on a well-formatted page.

CpSc 140: Introduction to Programming Principles (IT-R, IS-R, CS-R)
Project 1: Flash-Card Learning Game

Write a program that maintains pairs of words (or phrases) in an array or pair or arrays, one in English, the other in Spanish. The program allows the user to select which language is presented. During play, the program randomly selects a term from the appropriate array, displays it, and reads the user's response. The response is compared to the expected result, scored, and, if not correct, the correct answer is displayed. The user must be able to indicate when game play ends, at which time a final score is presented.

CpSc 150: Advanced Programming Principles (IT-E, IS-R, CS-R)
Project 1: Object Oriented Design

Create a program that simulates the operation of a vending machine. The user interacts with an interface that offers choices of food, accepts coins, and responds appropriately.

The program could offer a choice of language for the interface such as English or Spanish. Another option would be to offer a choice of ethnic foods such as Mexican or Italian or Classic American Fast Food.

CpSc 207: System Software and Architecture for End Users (IT-R)
Project 1: working with Basic UNIX commands

Create a file containing 10 English words each followed by a space character and then a one-word translation (using Alta Vista's Babel fish, Google's translation tools, or an actual bilingual dictionary) of that word into some other human language (extra credit for using non-Indo European languages). Write a shell script which accepts, as input, a word from either language and outputs its translation.

Discuss why this technique might not work for the general problem of translating text in one language to text in another language.

CpSc 210: Productivity Software (IT-E, IS-R)
Project 2

Prepare a PowerPoint presentation about a multinational software/hardware company based in Asia or Africa.

CpSc 217: Structured and Dynamic Web Programming (IT-R, CS-E)
Project 2: String handling and probability

First choose two languages to work with. One should be Indo-European, the other should be non-Indo European. The orthography of at least one should not be the Roman alphabet. Build a web page in which the user can type or paste a paragraph or more of text from either language. When done, the user can press a button which "analyzes" the text. The analysis consists of two tabulations: the frequency of occurrence of every glyph (character) including punctuation marks (In alphabets that distinguish between lowercase and uppercase letters, the two variants of the same letter should be considered the same so that "a" and "A" are identified) and the frequency of occurrence of every character pair (or digraph -- in which "ab" is not the same as "ba") is tabulated and presented on the page.

Finally, when a second button is pressed, new random text is generated and placed somewhere else on the page. The newly generated text should be generated with the probabilities of characters' occurrences matching the frequency of occurrence within the user's text. That is, if the occurrence of "e" accounts for 12% of the characters used, then the probability of generating an "e" should be 0.12

CpSc 236: Selected Computer Languages (IT-E, IS-E, CS-E)

All of the suggestions from CpSc 150 are appropriate for courses in other languages

CpSc 300: Challenges of Computer Technology (IT-R, IS-R)
Project 3

Why are there so few women and minorities in information technology? Use current literature to answer this question in an essay exam.

CpSc 301: Practical Computer Security (IT-E)
Project 2: What is a threat?

Different nations have different laws about freedom of speech, decency and obscenity, and privacy. Suppose you were sent by your employer to administer computing and network systems in a foreign country. Give a case study of how the laws of one particular country might affect your job.

CpSc 305: Introduction to Expert Systems (IS-E)
Project 3

Match up exchange students with host families based on cultural interests, language, ability/willingness to provide transportation, etc.

CpSc 311: Discrete Computational Structures (IT-R, IS-E, CS-E)
Project 1: History of Mathematics

List three major contributions to the history of mathematics that were contributed by persons neither from Europe nor the Americas.

CpSc 317: Scripting Languages (IT-R, IS-E)
Project 2: Mapping with Geographic Data

The map at http://granite.sru.edu/~ddailey/usmap.svg takes public data from the US Census Bureau and shades the 50 states based on any of 82 variables chosen by the user. Do the same thing for the states, provinces, or counties of another country where English is not the official language. Be sure to discuss issues involved in finding the data, its original sources and its accuracy.

CpSc 323: Database Systems (IT-R, IS-R, CS-E)

See Fig. 4.

CpSc 327: Systems Administration and Security (IT-R, IS-E)

See Fig. 2.

CpSc 343: File Processing (IT-E, IS-R)
Project 2

Design an HR database (set of files) that can be used by a company for employee management and healthcare management where the company allows domestic partners (and dependents). The system shouldn't "out" people just because they are using the benefits. Actually, this would be at least two "views" of the data.

CpSc 358: Simulation (CS-E)
Project 2: Best Fit and Bias.

Create best fit for data on table service for various ethnic and gender customers. The object would be to determine if service was fair among the various groups.

CpSc 365: Management Information Systems (IT-E, IS-R)
Project 2

Determine a way to recycle equipment to "needy people". How are they identified? How is the program sustainable? How does it help the corporation?

CpSc 370: Computer Organization (IS-E, CS-R)
Project 2

Write an assembly language with instructions appropriate for non-Western cultures.

CpSc 374: Algorithms and Data Structures (CS-R)
Project 1: Shortest-Path

For all of the major buildings on campus, develop a map (graph) that describes the path that wheelchair-bound person is able to take in traversing the campus. Weight each path segment based on a combination of difficulty of traversal and distance. Write a program that asks the user which building he or she is leaving and which he or she is traveling to. Determine the shortest path for the traversal.

CpSc 376: Programming Languages (CS-R)
Project 2

Select a language from the list below and prepare a presentation for the class that discusses the salient design decisions.

Designer	Language
Alain Colmerauer	Prolog
Ralph Griswold	Icon
Guido van Rossum	Python
Ole-Johan Dahl, Kristen Nygaard	Simula-67
'83:Jean Ichbiah, '95: Tucker Taft	Ada
Yukihiro Matsumoto	Ruby

CpSc 378: Theory of Computation (CS-R)
Project 3: The Game of “Choice”

Many still believe that sexual orientation is a matter of choice. It is well known among the ignorant, that proximity to those whose sexual preference varies from one’s own may cause one to question one’s sexuality or even to change. In Conway’s game of life, each cell has one of two states (dead or alive). In ours, each cell will have one of three states: gay, straight or questioning.

The program will read the state (rectangular map) from a file or allow the user to select cells interactively and assign a state. Each generation, including the initial is displayed. The rules for determining the next generation are as follows:

- If the cell is “straight” and 4 or more neighbors are straight, the cell stays the same.
- If the cell is “straight” and 2 or 3 neighbors are straight, the cell becomes “questioning”.
- If the cell is “straight” and 0 or 1 neighbors are straight, the cell becomes “gay”.
- If the cell is “gay” and 4 or more neighbors are gay, the cell stays the same.
- If the cell is “gay” and 2 or 3 neighbors are gay, the cell becomes “questioning”.
- If the cell is “gay” and 0 or 1 neighbors are gay, the cell becomes “straight”.
- If the cell is “questioning” and 4 or more neighbors are questioning or the number of neighboring gay cells equals the number of neighboring straight cells, the cell stays the same. Otherwise, the cell changes to the state of the maximum of gay and straight neighbors.

Submit with a DFA to show state changes.

Note: As implied above, this is not intended to model the real world. The “questioning” state is simply a contrivance for the program. We cannot replace “questioning” with bisexual, as we are not dealing with the gender of the neighbors. (And, a bisexual might well argue that he or she is not questioning his or her orientation.)

CpSc 413: Systems Analysis (IT-R, IS-R)
Project 2

Temp Office Personnel specializes in providing jobs for women with children. They match special skills to needs at a company and take into account the time needs of the mothers. A position is often filled by two, and potentially, more temps. All the skills have to match. Times need to correlate. And, they need a way to effectively transition work. Create a data model for such a system.

CpSc 423: Computer Networks (IT-R, IS-R, CS-R)
Project 3: Digital Divide

For each of the populated continents (Asia, Africa, North America, South America, Europe and Australia), identify at least three servers that will respond to ping and tracert in three different countries (assume Australia includes Tasmania, New Guinea, the Aru Islands and Raja Ampat Islands). For each of your servers, run ping tests to determine the minimum, average and maximum response times. Show the results using a graph. Use tracert to determine paths taken to each destination. Identify shared links in the paths and significant bottlenecks. Draw a graph showing the results.

CpSc 427: Interface Design (IT-R)
Project 2: Color perception and color terms

Write a paper on the neurological, perceptual and cultural issues involved in the use of colors on a web site. Include such topics as why women have more terms for colors than men and cross cultural differences in the emotive connotations of various colors.

CpSc 443: Software Project Management (IT-E, IS-R)

This assignment should be the next step of either of the CpSc 413 projects. Presentation on projects described in some selected set of papers (that would describe projects in Asia, Africa, etc.)

CpSc 450: Internship (IT-E, IS-E, CS-E)
Project 1: SRU Proprietary Report

After completing the internship, write a report that describes the company environment or culture as it pertains to multiculturalism. Describe how the culture could be improved while maintaining effectiveness and profitability

CpSc 456: Introduction to Computer Graphics (CS-E) Project 1: Avatar Creator

Many social networking sites allow the user to post an image which might be an avatar to represent them pictorially, some make more extensive use of avatars. For our purposes, we will assume a 2D, static avatar is the goal. Write a program that assists that user in creation of an avatar. Your program must take into account the preferred gender and race (skin tone) of the user. For extra credit, include height and weight preferences (for example, tall & slim).

CpSc 464: Principles of Concurrent Programming and Operating Systems (CS-R)

See Fig. 3.

CpSc 466: Compiler Design and Implementation (CS-E) Project 1

Use the biography of Vugranam Sreedhar as an introduction to a Compiler Design assignment.

In program analysis, compiler optimization, programming languages, security, business process, multicore Dr. Vugranam C. (VC) Sreedhar is currently a Research Scientist and a Project leader at IBM TJ Watson Research Center working in the area of Information Security. His Ph.D. thesis is in the area of program analysis and is entitled "Program Analysis Using DJ Graphs". Dr. Sreedhar has worked in several projects architecture, concurrency analysis, software quality, and embedded systems.

Jugranam C. (VC) Sreedhar is a lecturer with the ACM Distinguished Speaker Program on various topics including "Static Single Assignment Form and its Applications ".

ABSTRACT:

Static Single Assignment (SSA) Form is now a well accepted intermediate representation for compiler optimization. In this talk I will discuss the core ideas behind SSA form, related representations, and how they is used in improving program analysis techniques and in improving compiler optimizations. I will also discuss some of the key algorithms for constructing SSA form. I will then discuss some new applications of SSA in the area of typestate analysis and detecting security vulnerabilities.

CpSc 474: Computer Architecture (CS-R) Project 1: Architecture History

Find and report on accomplishments of computer scientists of specific gender and races One example of a woman is Admiral Grace Hopper, a woman who did research on early computers such as the Mark I and languages such as COBOL. Another example is an African, Philip Emeagwali, a Nigerian who made major contributions to the design of the Connection Machine. (www.math.buffalo.edu/mad/computer-science/emeagwali_philip.html)

CpSc 476: Artificial Intelligence (CS-E) Project 2

Create a program that helps the visually impaired learn about a specific topic such as the importance of color in Monet's paintings.

CpSc 478: Analysis of Algorithms (CS-E) Project 1

Use Djikstra's shortest path algorithm to perform a search of G=(V, E) where the vertices in the graph are cities in Africa and the edges of the graph are distances between African cities.

CpSc 488: Software Engineering (CS-R) Project 2: Project Design and Development

For the major course project, identify a female colleague or professional willing to act as customer-manager for each group. Students then develop the project iteratively with all "customer questions" directed to the customer-manager.