Online Resources for Teaching Programming to First Year Students

...

Alex Becheru & Costin Badică
Faculty of Automation, Computers & Electronics
University of Craiova, Romania

Outline

1. First year programming courses outline
2. Administrative support tools
3. Learning support tools
    a. For beginner students
    b. For intermediate students
    c. For advanced students
4. The end

First Year Programming Courses' outline

Computer Programming, 1'st Semester

- Language: ANSI C
- IO operations
- Conditional structures
- Repetitive structures
- Data structures / (vector, matrix)
- Operations with Strings
- Functions
- Recursivity

Programming techniques, 2'nd Semester

- Language: ANSI C + Python 3 (optional)
- Algorithm design and analysis
- Algorithm correctness and testing
- Sorting Algorithms
- Abstract Data types (Linked lists, Stacks & Queues)
- Graphs representation and traversal
- Dynamic programming

First Year Programming Courses' evaluation

Computer Programming, 1'st Semester

- Oral exam (50 %)
- Lab test on computers (25 %)
- Lab activity during the semester (25%)
  - Homeworks
  - Lab test on computers

Programming techniques, 2'nd Semester

- Written exam (40 %)
- Lab activity during the semester (30 %)
  - Homeworks
  - Lab test on computers
- Individual Project or Team Project (30 %)
  - Technical Documentation written in Latex
  - Doxygen generated documentation from code comments
  - Code solving specific problems

First Year Programming Courses' Info

Computer Programming, 1'st Semester

- 120 students
  - 60 Romanian teaching
  - 60 English teaching
- 4 assistants
- 1 teacher
- 14 labs (2h each)
- 14 courses (2h each)

Programming techniques, 2'nd Semester

- 120 students
  - 60 Romanian teaching
  - 60 English teaching
- 1 assistant
- 1 teacher
- 14 labs (2h each)
- 14 courses (2h each)
- 14 seminars (1h each)

Administrative support resources

Problem statement

- The University and Faculty does not enforce a common CMS.

Current situation

- Everybody uses whatever CMS is easier for them.
- Students are confused.
- Loss of time for teachers & students.

What do teachers need ?

- Tool to manage and give access to teaching materials.
- Tool to keep track of marks, attendance, etc.
- Tool to manage homeworks/handouts.

What do we think students need?

- Access to teaching materials
- Access to their current status (marks / attendance)
- Tool to manage homeworks/handouts.

Free solution from Google

- Collaborate on documents
- Share schedules/calendars
- Harness the full power of Gmail
- Work in group easier with Google Groups
- Easily create and manage sites via Google Sites
- Communicate and manage assignments

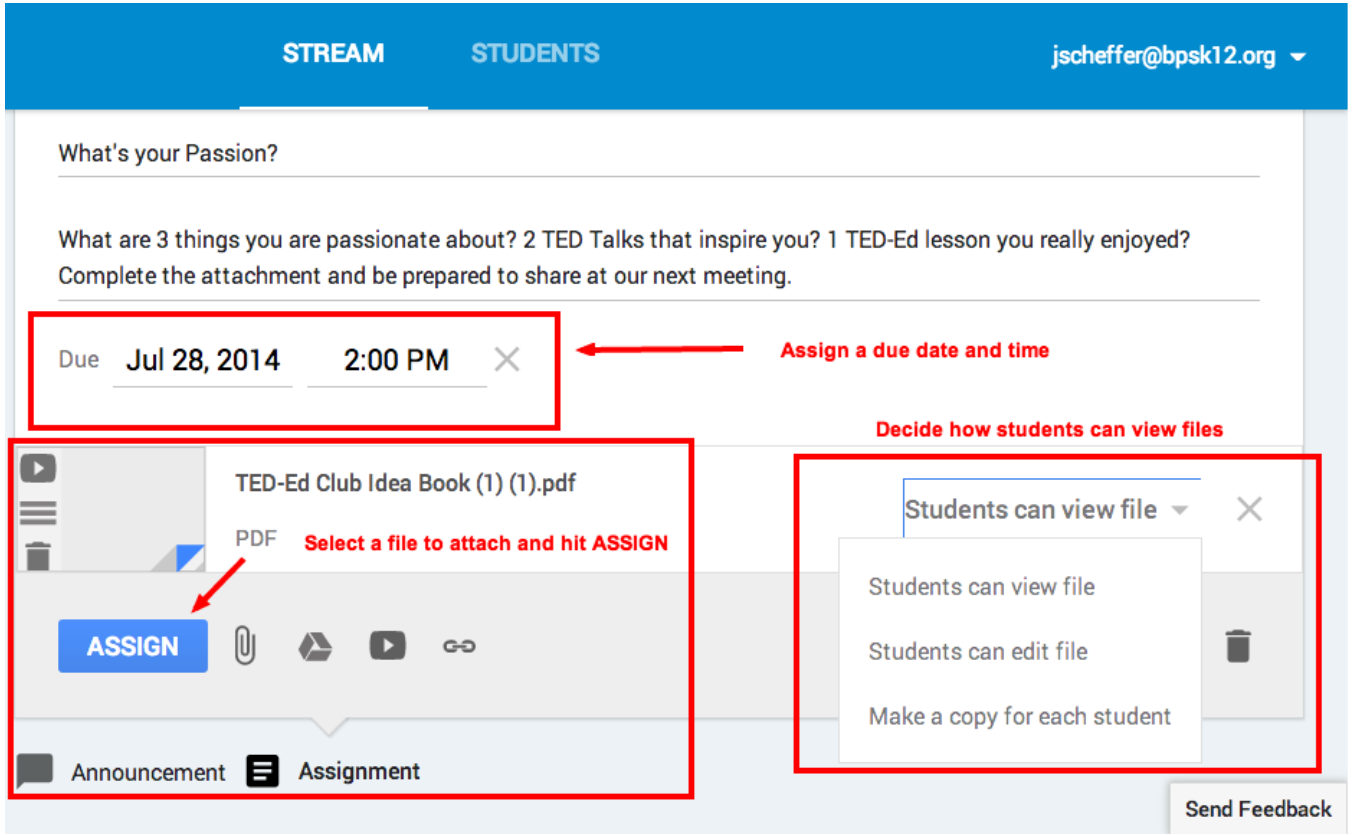
goo.gl/pySaku

Exploring Classroom:

Teacher Perspective

- Creating and customizing a new class
- Adding students
- Digital workflow - assignments
- Digital communication - announcements & email
- Integration with Google Drive

Assignment Features



Assignment: Basic Italian Vocabulary

DUE AUG 12

RETURN | EMAIL | FOLDER | DOWNLOAD

Student	Status	Grade	Points 100
Zach Yeskel Well done, Zach. I owe tell you have	DONE Aug 7	100/100 *Not Returned*	
Ben Schrom Send a note	DONE	No Grade	
Jen Miller Thanks, Jen. Great job. Moon is	RETURNED Aug 7	98/100	
Sheryl Jackson Send a note	NOT DONE	No Grade	
Mike McCollum Send a note	NOT DONE	No Grade	
Matt Thomas Well done, Matt!	RETURNED Aug 7	92/100	
Elizabeth Frank Send a note	DONE	86/100 *Not Returned*	
Brittany McGinnis Send a note	DONE	No Grade	
Chat Atapattu Send a note	NOT DONE	No Grade	

Send Feedback

Exploring Classroom:

Student Perspective

- ❑ Joining a new class
- ❑ Digital communication - discussion & email
- ❑ Digital workflow - creating & submitting assignments
- ❑ Integration with Drive
- ❑ Opportunity to integrate digital citizenship

Manage Homeworks/ Handouts

- Html web form to upload files to Google Drive
- Easy to personalize script
- The uploader can download/view the file uploaded
- Only the teacher has access to all files

http://goo.gl/6GRl5p

Sharing Code

- Codepad is an online compiler/interpreter
- A simple collaboration tool
- Codepad will give you a short URL you can use to share your code

CODEPAD.ORG

http://codepad.org/

Learning support resources

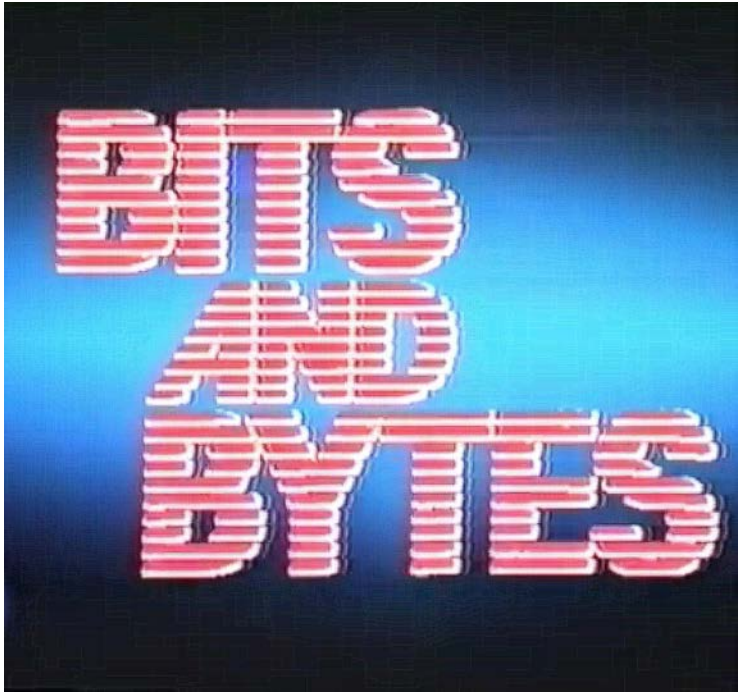
Problem statement

- Students learn differently due to their background and personality.
- First year student’s prior knowledge of programming varies.
- Students have to be kept in the loop, they should not be presented with too advanced/basic knowledge.

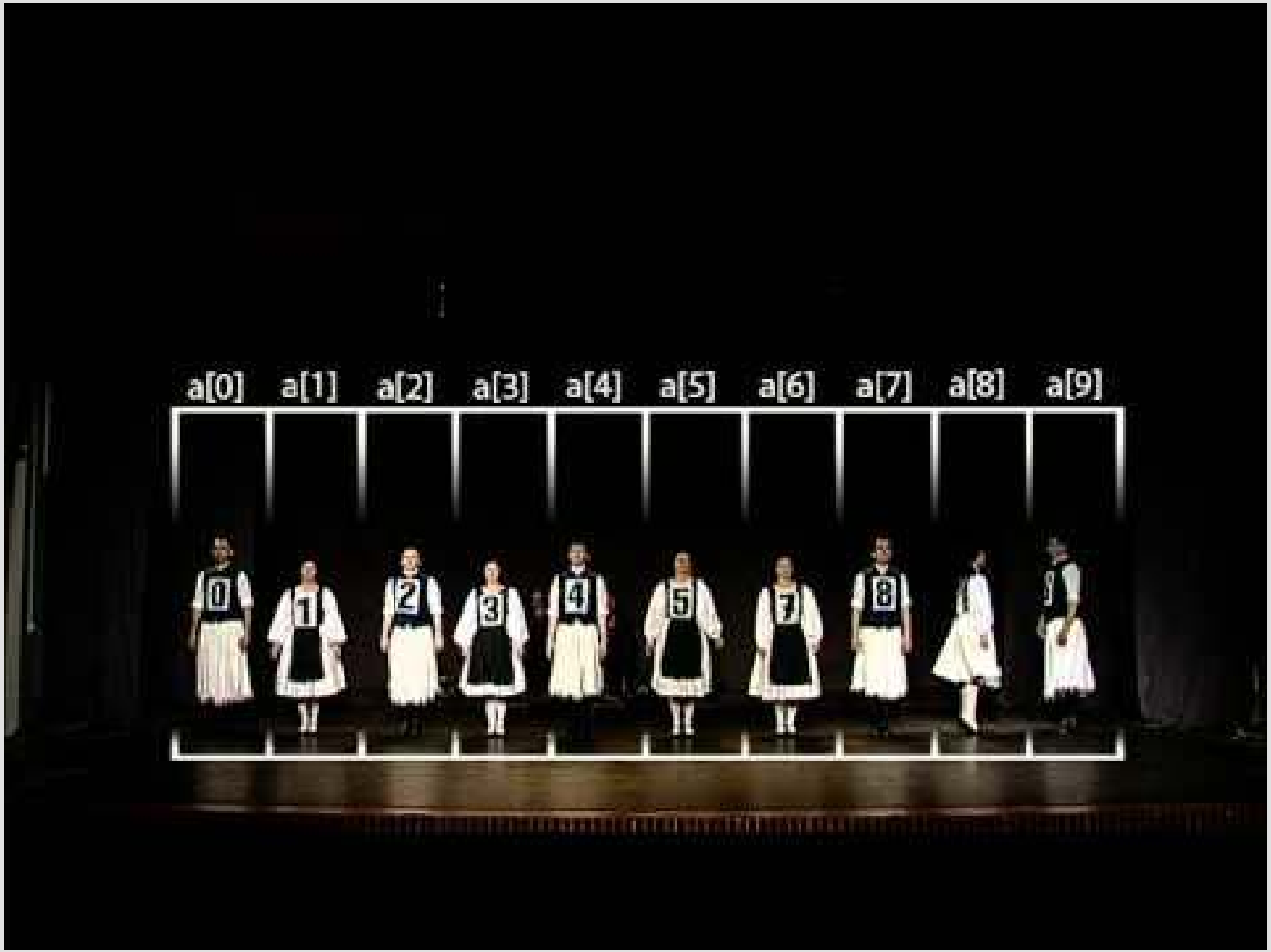

NOOB

Bits and Bytes TV Series

- How programs work
- File and data management
- Communication between computers
- Computer languages
- Computer assisted instructions
- Simulations and Games
- Computer Graphics
- Computer Music



https://goo.gl/AcGRUZ


a[0]
a[1]
a[2]
a[3]
a[4]
a[5]
a[6]
a[7]
a[8]
a[9]

Youtube Course about C

- Binary tutorial
- Hexadecimal tutorial
- Numeric overflow
- Why do we need to learn pointers?
- Changing data using bitmasks

Free Online Programming

Course in C for Beginners

Carl Herold

https://goo.gl/Brxfw4

Learn C by doing

- 4.c: printf output
- 12.c: nested for loops
- 35.c: multiple files compiled together
- 46.c: Arrays of Structures
- 59.c: pointers to structures
- 78.c: register const storage qualifiers

88 C Programs

JT Kalnay

https://goo.gl/yUT0Ss

Algorithms for beginners

- Lecture Videos & Notes
- Handouts
- Problem statements
- Topics:
  - Greedy algorithms
  - Dynamic programming
  - ...

Intensive post-baccalaureate program in Computer Science based on the undergraduate course of study at MIT.

http://aduni.org/courses/algorithms/index.php?view=cw

Algorithms Visualisation

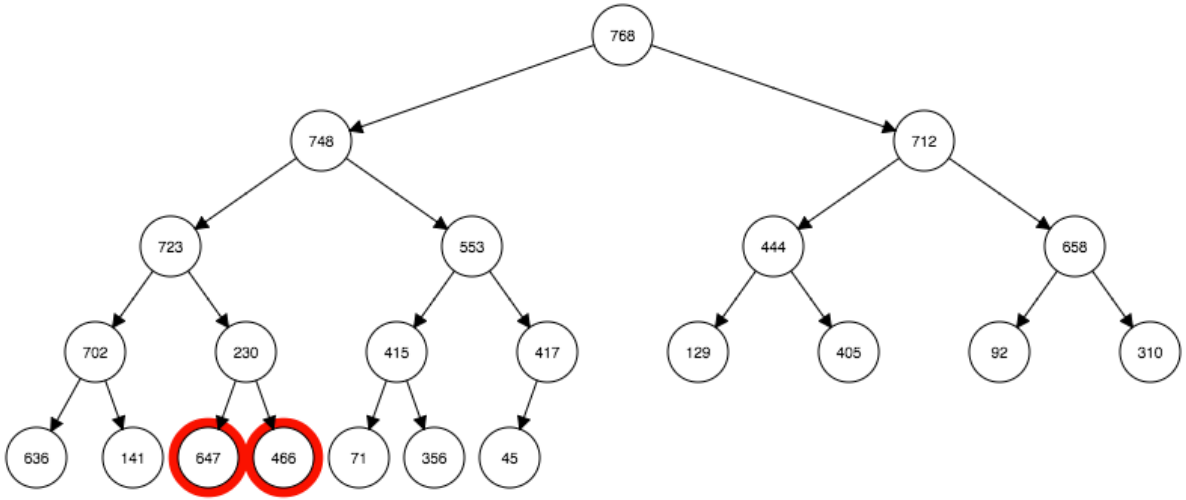
- Basics: stacks, queues, linked lists, ...
- Recursion: factorial, n-queens, ...
- Indexing: B+ Trees, Red-Black Trees, ...
- Sorting: Shell Sort, Radix Sort, Heap Sort
- Heap-like Structures: Skew, Leftlist, ...
- Graph Algorithms: DFS, Kruskal, Prim, ...
- Dynamic Programming
- Geometric Algorithms
  - 2D Rotation and Scale Matrices
  - 2D Rotation and Translation Matrices
  - 2D Changing Coordinate Systems
  - 3D Rotation and Scale Matrices
  - 3D Changing Coordinate Systems
- Others: DIsjoint Sets, Huffman Coding, etc.

https://www.cs.usfca.edu/~galles/visualization/Algorithms.html

Heap Sort

Randomize Array | Heap Sort

768	748	712	723	553	444	658	702	230	415	417	129	405	92	310	636	141	647	466	71	356	45	777	819	851	901	915	953	960	971	980
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30



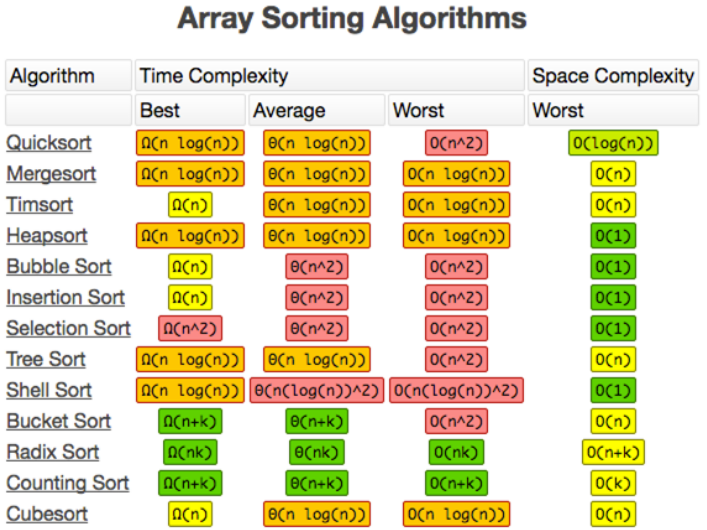
Animation Running

Skip Back | Step Back | Pause | Step Forward | Skip Forward | Animation Speed | w: 1000 h: 500 | Change Canvas Size | Move Controls

Algorithm Visualizations

Algorithms Complexity

- Common data structure alg: array, B-Tree, AVL, KD-Tree, ...
- Array sorting: Timsort, Quicksort, Radix, …
- Graph data structure: incidence list, adjacency list, …
- Heap Data Structure: Binary, Fibonacci, …
- Graph alg: A*, Topological sort, …



Array Sorting Algorithms

Algorithm	Time Complexity			Space Complexity
	Best	Average	Worst	Worst
Quicksort	Ω(n log(n))	θ(n log(n))	O(n^2)	O(log(n))
Mergesort	Ω(n log(n))	θ(n log(n))	O(n log(n))	O(n)
Timsort	Ω(n)	θ(n log(n))	O(n log(n))	O(n)
Heapsort	Ω(n log(n))	θ(n log(n))	O(n log(n))	O(1)
Bubble Sort	Ω(n)	θ(n^2)	O(n^2)	O(1)
Insertion Sort	Ω(n)	θ(n^2)	O(n^2)	O(1)
Selection Sort	Ω(n^2)	θ(n^2)	O(n^2)	O(1)
Tree Sort	Ω(n log(n))	θ(n log(n))	O(n^2)	O(n)
Shell Sort	Ω(n log(n))	θ(n(log(n))^2)	O(n(log(n))^2)	O(1)
Bucket Sort	Ω(n+k)	θ(n+k)	O(n^2)	O(n)
Radix Sort	Ω(nk)	θ(nk)	O(nk)	O(n+k)
Counting Sort	Ω(n+k)	θ(n+k)	O(n+k)	O(k)
Cubesort	Ω(n)	θ(n log(n))	O(n log(n))	O(n)

http://bigocheatsheet.com/

Git and Github Course

- Keep track of multiple versions of a file
- Track bugs by reverting to previous working versions of a file
- Seamlessly collaborate with other developers on a project

https://www.udacity.com/course/how-to-use-git-and-github--ud775

Coding Style

- Tabs are 8 characters, and thus indentations are also 8 characters.
- “If you need more than 3 levels of indentation, you're screwed”
- The limit on the length of lines is 80
- “If you have some random integer loop counter, it should probably be called "i". “
- “You know you're brilliant, but maybe you'd like to understand what you did 2 weeks from now. ”

Linux Kernel indentation

Linus Torvalds

https://goo.gl/nZbVKd

LaTeX online introduction

- Introduction to LaTex
- Producing simple documents using LaTex
- Producing Mathematical Formulae using LaTeX
  - Greek Letters
  - Matrices and other arrays in LaTeX
  - …
- Further Features of LaTeX
  - Lists
  - Tables
  - …

University of Dublin
School of Mathematics

David R. Wilkins

http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/

LaTeX online IDE

- Collaboration
- Ease of Use
- Document history
- Dropbox sync
- Templates
- Easy to use LaTeX documentation

https://www.sharelatex.com/

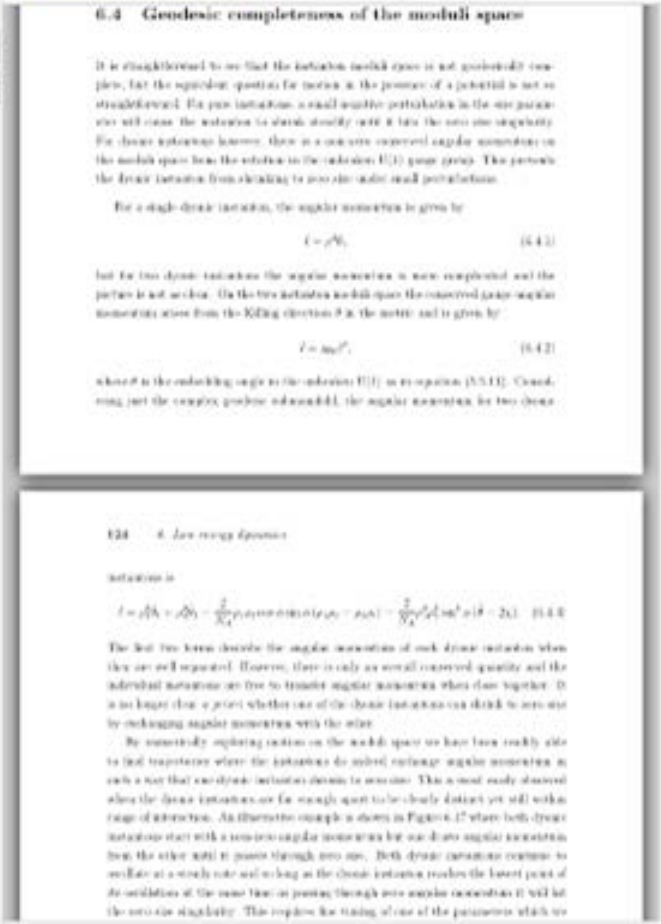
figures
sections
instantons
introduction.tex
low-energ...
moduli-space.tex
m2-branes
polytopes
abstract.tex
conclusion.tex
introduction.tex
bibliography.bib
dmathesis.cls
logo.pdf
main.tex
preamble.tex
utphys.bst

```
\section{Geodesic completeness of the moduli space} \label{sec:geodesic completeness of the moduli space}

It is straightforward to see that the instanton moduli space is not geodesically complete, but the equivalent question for motion in the presence of a potential is not so straightforward. For pure instantons, a small negative perturbation in the size parameter will cause the instanton to shrink steadily until it hits the zero size singularity. For dyonic instantons however, there is a non-zero conserved angular momentum on the moduli space from the rotation in the unbroken $\U(1)$ gauge group. This prevents the dyonic instanton from shrinking to zero size under small perturbations.

For a single dyonic instanton, the angular momentum is given by
\begin{equation}
l = \rho^2 \dot \theta,
\end{equation}
but for two dyonic instantons the angular momentum is more complicated and the picture is not as clear.
On the two instanton moduli space the conserved gauge angular momentum arises from the Killing direction $\theta$ in the metric and is given by
\begin{equation}
l = g_{\theta p} \dot z^p,
\end{equation}
where $\theta$ is the embedding angle in the unbroken $\U(1)$ as in equation \eqref{eq:polar coordinate metric}.
```

Recompile



C for beginners

- Lecture Slides
- Project assignments
- Assignments
- Topics:
  - Multithreading and concurrency
  - Garbage collection
  - ...

MIT OPENCOURSEWARE
Practical C programming

http://goo.gl/ZRkDN6

Python Course for beginners

- 13 Hours Estimated Course Time
- 9 Projects
- 9 Quizzes
- Online Console

https://www.codecademy.com/learn/python

Coding Style

- All loops must have a fixed upper-bound
- No function should be longer than what can be printed on a single sheet of paper
- The return value of non-void functions must be checked by each calling function

NASA's 10 rules for Developing Safety Critical Code

Gerard J. Holzmann (JPL)

http://goo.gl/hURmHp


WILL CODE
HTML For
FOOD
PLEASE HELP

becheru@gmail.com