CSCI 160 — Introduction to Problem-solving & Programming

Jeremy Iverson

College of Saint Benedict & Saint John's University

logistics

- instructor(s)
  - Jeremy Iverson (`jiverson002@csbsju.edu`)
  - PENGL 213, (320) 363-3083
  - office hours: MW 12:45 - 13:45
    T 11:30 - 12:30
- textbook
  - *Big Java: Early Objects*, 6 Edition, Horstmann
- website
  - `https://csbsju.instructure.com/courses/7691`
- labs
  - PENGL 218, TR 01A 08:00 - 09:20

I prefer to be called Jeremy
Encourage questions right away
Emphasize the importance of the Canvas site for finding information about the class

course objectives

- write well-designed Java programs for small-scale problems
- identify appropriate object classes for solving a given problem
- write good documentation for programs

Will not be asked to write full-scale applications

- Emphasizes programming fundamentals, special attention to object-oriented software design (ptyon and visual basic are both examples)
- Problem-solving

course content

- introduction to Java
- using methods
- using objects
- decisions
- iterations
- arrays
- designing and implementing classes
- inheritance
- interfaces and polymorphism
- file i/o
- exceptions
- recursion
- searching
- sorting
- collections
- advanced topics

Ask students to point out some of the topics which are covered in 130/150
Point out that each of these bullets corresponds to one of the “modules” on Canvas

labs

- attendance
- prelabs

- I will be sure to remind for the first couple labs, after that
- Your responsibility to go on N drive to course folder to see if there is a prelab
- I will try to remind, but it is student's responsibility
- Amounts to two (roughly 30 min) assignments per week
- Labs are due Tuesday following the lab session that they are assigned

assignments

- do your own work
- use each other as resources only

Should be student's own work, may work together but must individual solutions

Want to encourage using each other as resources, but do not want some students to rely on other students for answers

quizzes and exams

- six (6) in-class quizzes (closed book / note)
- three (3) lab exams (open book / note)
- two (2) in-class exams (closed book / note)
- one (1) in-class final exam (closed book / open note) [cumulative]

Make-up exams are possible under certain circumstances, but no make-up quizzes

office hours

- stop by anytime
- my calendar

Mention outlook calendar & my home page
For those unfamiliar with Outlook meetings, then they should schedule another way and we will go over this in meeting
Mention not generally availability in the afternoons

teaching philosophy

Your job is to empower those you teach; when you do for them what they should be doing for themselves, you create dependency rather than empowerment.

It is easy to give in to the frustration that results from seeing amazing possibilities for the people you are teaching, and you want it more for them than they want it for themselves.

Don't give in to that frustration!

- Based on passage from "Resisting Happiness" by Matthew Kelly

What do you think I mean by dependency?

Dependency — relying on the instructor, TA, or fellow student for original thought or deeper understanding (in context of computer science) instead of oneself

expectations

- prepare for class (do the reading!)
- participate in class
- be respectful

- Don't just read, by try your best to make sense of the material
- Emphasize that students are not just recipients of my knowledge — they can shape the direction of the course
- Encourage students to stop and think before replying.
- Don't be discouraged if comprehension is not apparent right away, but don't be complacent
- Seek help (TAs and me)
- Students can expect the same things from me.

questions?

notecards

- given name
- preferred name and pronunciation
- anything you would like me to know about you
- at least one of the following:
  - reason for taking the class
  - what you are hoping to learn from the class

Java program

- open Lecture1.java
- follow the instructions

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177