

Test #1 Overview

Material covered

- Chapters 2 - 4 in Introduction to Programming with C++, 3rd Edition by Liang
- Homework and MyProgramming Lab Assignments: Ch2, Ch3, Ch4

Format

- No books, no notes, no computers
- Types of problems includes:
 - Determining the output of programs on the test
 - Some T/F, multiple-choice, short answer, etc.
 - Writing programs or instructions to accomplish specified tasks.
- Very detail-oriented. Be prepared!

Items provided on the test (also see documents on web site)

Tables of ASCII Codes and Operator Precedence – also attached

Items provided on the test (also see documents on web site)

Tables of ASCII Codes and Operator Precedence

Sample Test – See sample test on course web site.

Chapter 2 - Elementary Programming

Basic features of programs:

- using namespace std;
- #include <iostream>
- form of main function, return 0
- use of cin for keyboard inputs and cout for display outputs

No questions related to compilers, projects, etc.

Declaring variables (identifiers)

Rules for identifier names

Reserved words – only know those that we have used

Assignment statements

Data types – int, float, double, char, and bool only. Do not need to know # bytes or ranges.

Constant variables

Mixed-type operations

Type casting - use of static_cast<int>(), static_cast<double>(), static_cast<float>(), etc

String constants (string literals)

- use double quotes
- know the escape sequences \t, \n, \v, \", \\, \'
- used in Chapter 2 in cout statements, such as cout << "Hello World!" << endl;

Character (char) variables

- use single quotes
- assigning values - Example: char c1 = 'A'; (same as char c1 = 65;)
- treated as integers based on ASCII code (table provided)
- know the escape sequences \t, \n, \v, \", \\, \'

Algebraic expressions, mixed-type operations, precedence

Increment and decrement operators (including pre-increment, post-increment, etc)

Compound assignment operators (shortcut operators): +=, -=, /=, *=, %=

Precedence of operators – table provided

Miscellaneous

Chapter 3 – Selection Structures

if structures

- Various forms (if, if else, if else if, etc)
- Rational expressions (6 operators: ==, >=, >, !=, <=, <)
- Logical expressions (3 operators: &&, ||, !) - know their truth tables
- Precedence of operators
 - table provided
 - evaluate conditional statements involving arithmetic, relational, and logical operators
- Nested if structures
- Using arithmetic expressions for conditional statements
 - Evaluation of logical expression yields True (1) or False (0)
 - Non-zero arithmetic expression treated as True, zero as False
 - Example: if(x)

Conditional expressions (ternary operators): Example: $y = (x > 2) ? 10 : 20;$

Switch structures

- int or char expressions only
- default case
- break statements
- drop-through behavior & grouping cases

Bool data type

Miscellaneous

Chapter 4 – Math functions, characters, strings, and I/O manipulators

Math functions

- use cmath header
- sin, cos, tan, asin, acos, atan, pow, sqrt, ceil, floor, abs, log, log10, max, min

Characters

- compare using relational operators and ASCII codes

Strings

- use string header (class)
- string variables
- assigning strings to variables – use double quotes
- string operators: + (concatenation) and +=
- string functions: .at(), .length(), .size()
- accessing string elements using stringname[]
- compare using relational operators, ASCII codes, and lexicographical order (the order used in a dictionary) Ex: if (Answer == “Yes”) Ex: If (“Jones” > “Johnson”)

Formatted output

- setprecision(n), setw(n), fixed, scientific, setfill('*'), showpoint, left, right
- use iomanip header

Miscellaneous

Precedence of Operators

Operator Symbol	Operator Name	Direction	Precedence (1 = highest)
()	Parentheses	L to R	1
++, --	Post-increment	L to R	2
++, --	Pre-increment	R to L	3
!	Logical NOT	L to R	3
+, -	Positive, negative	L to R	3
*, /, %	Multiplication, division	L to R	4
+, -	Addition, subtraction	L to R	5
<=, >=, >, <	Relational operator	L to R	6
=, !=	Relational operator	L to R	7
&&	Logical AND	L to R	8
	Logical OR	L to R	9
+=, -=, *=, /=, %=	Compound assignment	R to L	10
=	Assignment	R to L	10

Table of ASCII Codes

Character	ASCII decimal equivalent	Character	ASCII decimal equivalent	Character	ASCII decimal equivalent
\a	7	<	60	_	95
\b	8	=	61	`	96
\t	9	>	62	a	97
\n	10	?	63	b	98
\v	11	A	65	c	99
\f	12	B	66	d	100
\r	13	C	67	e	101
space	32	D	68	f	102
!	33	E	69	g	103
"	34	F	70	h	104
#	35	G	71	i	105
\$	37	H	72	j	106
&	38	I	73	k	107
'	39	J	74	l	108
(40	K	75	m	109
)	41	L	76	n	110
*	42	M	77	o	111
+	43	N	78	p	112
,	44	O	79	q	113
-	45	P	80	r	114
.	46	Q	81	s	115
/	47	R	82	t	116
0	48	S	83	u	117
1	49	T	84	v	118
2	50	U	85	w	119
3	51	V	86	x	120
4	52	W	87	y	121
5	53	X	88	z	122
6	54	Y	89	{	123
7	55	Z	90		124
8	56	[91	}	125
9	57	\	92	~	126
:	58]	93		
;	59	^	94		