Unit 3, Lesson 2
Data Types, Arithmetic, Variables, Input, Constants, & Library Functions

Mr. Dave Clausen
La Cañada High School

Vocabulary

- Variable- A variable holds data that can change while the program is running
- Constant- A constant is used to store data that remains the same throughout the program's execution
- Integers- Integers include the whole numbers and the negative numbers (no decimal points).
- Floating point numbers- Floating-point numbers are real numbers (have decimal points)
- Data type- Data type is a type of variable that best fits the nature of the data itself

Integer Data Types

- When you are working with either positive numbers, negative numbers, or zero you should use integer data types for you variables
- Several integer data types are available in C++ (integer data types can vary by the complier)

Simple Data Types: Integers

- Type int

  represent integers or whole numbers

  Some rules to follow:

  - Plus signs do not need to be written before the number
  - Minus signs must be written when using negative numbers
  - Decimal points cannot be used
  - Commas cannot be used
  - Leading zeros should be avoided (octal or base 8 numbers)
  - limits.h int_max int_min

Simple Data Types: Double

- Type double
  - used to represent real numbers
  - many programmers use type float, the AP Board likes the extra precision of double
  - avoid leading zeros, trailing zeros are ignored
  - limits.h, float.h
    - dbl_max, dbl_min, dbl_dig

Simple Data Types: Characters

- Type char

  used to represent character data

  - a single character which includes a space

  must be enclosed in single quotes eg. ‘d’

  Escape sequences treated as single char

  - ‘\n’ newline
  - ‘\’’ apostrophe
  - ‘\”’ double quote
  - ‘\t’ tab

Simple Data Types: Booleans

- Type bool
- A Boolean Variable is a variable that can have only two possible values:

  true

  false
- Boolean variables help us represent answers as yes, no, true, or false.

Primitives

- Simple data types such as:

  int

  float

  double

  char

  bool

  are called Primitive data types, or Primitives for short.

Integer Arithmetic

- + Addition
- - Subtraction
- * Multiplication
- / Quotient (Integer Division)
- % Remainder (Modulus)

$$Divisor \overline{)\, Dividend} \;\; \begin{matrix} Quotient \\ + \\ \end{matrix} \;\; \frac{\text{Remainder}}{Divisor}$$

Integer Order Of Operations

- Expressions within parentheses
  - nested parentheses: from inside out
- * (multiplication), % (modulus), / (division)
  - from left to right
- + (addition), - (subtraction)
  - from left to right

Integer Arithmetic (Examples)

(3-4)*5 =	-5
3 * (-2) =	-6
17 / 3 =	5
17 % 3 =	2
17 / (-3) =	-5
-17 % 7 =	-3
-42+50%17=	-26

Integers

- Stored as binary numbers inside the computer.
- Integers produce exact answers
- Int_Min and Int_Max

  -32,768 and 32,767
- Integer Overflow

  a number is too large or too small to store

  no error message

  unpredictable value

Real Number Arithmetic

- Type double:

- + Addition
- - Subtraction
- * Multiplication
- / Division

Real Number
Order Of Operations

- Expressions within parentheses
  - nested parentheses: from inside out
- * (multiplication), / (division)
  - from left to right
- + (addition), - (subtraction)
  - from left to right

Real Number Arithmetic (Examples)

$2.0 * (1.2 - 4.3) = -6.2$

$2.0 * 1.2 - 4.3 = -1.9$

$-12.6 / (3.0 + 3.0) = -2.1$

$3.1 * 2.0 = 6.2$

$-12.6 / 3.0 + 3.0 = -1.2$

Real Numbers

- Bring Calculators to check your math
- Are stored using binary numbers
- Round Off Error

  1.0/3.0 = 0.3333…….
- Underflow

  very small numbers may be stored as zero

  0.00000000000000000123 stored as zero

Real Numbers

- Representational errors

  precision of data reduced because of the order in which operations are performed

  (-45.5 + 45.6) + 0.215 = 0.315

  - 0.1 + 0.215 = 0.315

  -45.5 + (45.6 + 0.215) = 0.3

  - if three digits of accuracy are the computers limit
  - 45.6 + 0.215= 45.815 or 45.8
  - -45.5 + 45.8 = 0.3

Real Numbers

- Cancellation Error

  lost data due to differences in the precision of operands

  2 + 0.0005 = 2.0005 but only 2.00 if 3 digits of precision

  If possible, add all small numbers before adding to a larger number

  Real Overflow: trying to store very large numbers

Identifiers

- Must start with a letter of the alphabet or underscore _
  (we will not use underscores to start identifiers)
- after your first letter, you can use numbers or letters or underscore characters (no spaces)
- length determined by compiler

  Borland C++ unlimited
- common use is to name variables & constants

Variables and Constants

- Computers store data in many complex arrangements called data structures.
- Most data is stored in either variables or constants. Variables hold data that can change while the program is running. Constants are used to store data that remains the same throughout the program's execution.
- Variables must be declared before they are used. Variables should also be initialized to clear any random values that may be in the memory location. When a variable is declared, it must be given a legal name called an identifier.
- Constants are declared in a way similar to variables. The const keyword tells the compiler that the data is a constant. The constant must be assigned a value in the declaration statement.

Variables and Constants 2

- Memory Location

  storage cell that can be accessed by address

- Variable

  memory location, referenced by identifier, whose value can be changed during a program

- Constant

  Symbol whose value can’t be changed while the program is running.

Naming Variables

- The names of variables are typically referred to as identifiers
- When naming variables, use descriptive names and consider how the might help the programmer recall the variable's purpose
- For example, the circumference of a circle should be stored in a variable named circumference.

Software Engineering

- Self-documenting code

  Code that is written using descriptive identifiers

- Always use descriptive variable names and constant names

  Remember: aim for 5 to 15 characters

  Borland C++ for Windows both can handle 32 characters for identifier names

Declaring and Naming Variables

- Indicating to the complier what type of variable you want and what you want to name it is called **Declaring** the variable
- You must declare a variable before you can use it
- The C++ statement declaring a variable must include the data type followed by the name you wish to call the variable and semicolon
- **Programming Style**: variable names are all lower case. If more than one word, use an underscore between the two words as spaces are not allowed in identifier names.

Declaring Variables

- Variables must be declared before they can be used.
- For example:

```
int sum;
double average;
char menu_choice;
bool is_prime;
```

Initializing Variables

- The complier assigns a location in memory to a variable when it is declared
- However, a value already exists in the space reserved for your variable
- To initialize a variable, you simply assign it a value
- The assignment operator (=) is used to assign a value to a variable

Declaring and Initializing Variables

- Variables can be declared and assigned an initial value in the same step:
- For example:

```
int sum = 0;
double average = 0.0;
char menu_choice = ‘q’;
bool is_prime = false;
```

Variable Example

```
#include <iostream.h>  // necessary for cout command
    int main()
    {
        int i;   // declare i as an integer
        i = 2;  //  initialize i to 2
        cout  << i  << endl;
        return 0;
    }
```

Assignment Statements

- A Method of putting values into memory locations

  <variable name> = <value>;

  <variable name> = <expression>;
- Assignment is made from right to left
- Constants can’t be on left side of statement
- Expression is a Constant or variable or combination thereof

Assignment Statements

- Values on right side not normally changed
- variable and expression must be of compatible data types (more later)
- Previous value of variable discarded to make room for the new value
- For now, char, int, and double are compatible with each other

Assignment Examples

- score1 = 72.3;
- score2 = 89.4;
- score3 = 95.6;
- average = (score1 + score2 + score3) / 3.0

  why not divide by 3 instead of 3.0?

Compound Assignments

- "Short hand" notation for frequently used assignments (We will not use these for readability of our programs.)

Short hand	Longer form
x += y	x = x + y
x -= y	x = x - y
x *= y	x = x * y
x /= y	x = x / y
x %= y	x = x % y

Sample Program

Here is a program that prints data about the cost of three textbooks and calculates the average price of the books:

Books.cpp

Character Data

- Type char

  each char is associated with an integer value

- Collating sequence

  order of character data used by the computer

- Character set

  the character list available

  ASCII (American Standard Code for Information Interchange) on our systems

ASCII Code

	0	1	2	3	4	5	6	7	8	9
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT
1	LF	VT	FF	CR	SO	SI	DLE	DC1	DC2	DC3
2	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS
3	RS	US	SP	!	"	#	$	%	&	`
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	'	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	DEL		

Full ASCII Code Chart

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Constants

- Symbolic constant: PI
- Literal constant: 3.14
- Constant declaration section

  after Preprocessor Directives

  before type definition section, program heading, and the int main( ) function.

  Literal constant is assigned to symbolic constant

  - const double PI = 3.14;
- **Programming Style** for constants is ALL_CAPS

Rationale for using Constants

- Programs are easier to read
- Easier to change values that are currently fixed but subject to change in the future

  i.e. STATE_TAX_RATE

  change one line, rather than searching every line
- Programs more reliable

  fewer chances for typos

  compiler will “catch” named constant typos

Library Constants

- What are the ranges from minimum to maximum for the types char, int, double, etc.?

  # include <limits.h>;

  # include <float.h>;
- Varies by computer system
- Here is a test program to display the values

  sizes.cpp

Input

- Cin (pronounced see-in)

  gets data from keyboard, the standard input stream

  extractor operator >>

  - obtain input from standard input stream and direct it to a variable (extract from stream to variable)

  inserter operator <<

  - insert data into standard output stream

  EGG ILL

  - Extractor Greater Greater, Inserter Less Less

Input

- Data read in from keyboard must match the type of variable used to store data
- Interactive Input

  enter values from keyboard while the program is running

  cin causes the program to stop and wait for the user to enter data from the keyboard

  prompt the user for the data (user friendly)

Input: Sample Programs

No prompt for any of the data values:

input.cpp
input.txt

One prompt for each data value (preferred)

triples.cpp
triples.txt