







]	Projects inVisual C++	
File \rightarrow New \rightarrow Project	t or <i>Ctrl+Shft+N</i> – to create a new project	
Choose : Win32 (Console Application	
Name: supply pr	oject name, e.g. welcome	
Location: cho	ose folder to save project	
In Application Se	ttings choose	
File $\rightarrow Open \rightarrow Project$	ct or Ctrl+Shft+O – to open an existing project	
File \rightarrow New \rightarrow File or	Ctrl+N	
– to c	reate a new file without adding to project	
File $\rightarrow New \rightarrow File$ or	Ctrl+O	
- to o	pen an existing file without adding to project	
Projects →Add New I	tem or Ctrl+Shft+A	
- to cr	eate a new C++ source file <u>and</u> add it to project	
In Templates cho	ose: C++ File (cpp)	
Name: supply fil	e name, e.g. hello	
Location: cho	ose folder to save file	
Projects →Add Existin - to in	<i>ng Item</i> or <i>Alt+Shft+A</i> clude existing file into project	
To exclude a file from	a project (the file is NOT deleted): right click on the file	e and select <i>Remove</i>
1/15/2010	MET CS 563Spring 2010 2. Programming Fundamentals	5

	D (a build (a subjict the subject)
Ctr + Snft	B - to build / complie the project
Build $\rightarrow B$	uild Solution
~	
<i>Ctr</i> + <i>F5</i> - t	o run the project outside debugger
or	
Debug $\rightarrow S$	start Without Debugging
F5	- to run the project intside debugger
or	
Debug $\rightarrow S$	Start





Valid ar	nd Invalid Identifiers
Valid or not? Why?	
sum abc	✓ ✓
AnneX-2 error% lst-approximation al. bl	no: '-' is special character no: '%' is special character no: '-' is special character
\$sum totalProduct _simple s2	 no: '\$' is special character and good style but may cause confusion but may cause confusion
win?gate	<i>no:</i> '?' is special character
1/15/2010 MI 2. Pr	ET CS 563Spring 2010 9 ogramming Fundamentals

	Naming Rules an	d Styles
Case sensi different cl	<i>tive</i> : upper and lower cas naracters, thus	e of the same letter are
Count and	count refer to different	objects.
Convention	n: begin object name with	h lower case letter, e.g.
speed	point sum	
Convention composite	n: use upper case letters i names more readable, e.	n the name to make g.
totalSum	hourlyPay	(preferred to
total_sum	hourly_pay)







Computation	s are done by evaluating expressions, e.g.
count + 5	
speed*time	
The values of place in mem	the above expressions will be lost if not saved to a specific ory. This is done through an assignment statement, e.g.
distance =	speed*time;
More general	ly
<identifier></identifier>	= <expression>;</expression>
Another type	of statement
cin >> speed	;
The expression interest is the the object sp	on is evaluated, but its value is not interesting. What is of <i>side effect</i> it produces that consists in changing the value of ed to whatever is entered from the keyboard.



	Pre	ecedence and	Associativ	vity	
Problem: In	n what orc	ler should opera	itors be execu	ted?	
• Precedence operation is relative to e	: In an e determine ach other.	xpression conta ed by rules esta	ining <u>differen</u> blishing a preo	t operators the cedence of the	order of operator
Highest ←	()	++ 	* / %	+ -	→ Lowest
• Associativit of execution or from righ	ty: In a sent is define to to left (•	equence of opera ed either from le ←), called <i>right</i>	ators of the <u>sa</u> ft to right (\rightarrow) <i>-associative</i>	me precedence), called <i>left-as</i> .	the order sociative,
<u>a*b*</u>	c →	<u>a+b</u>	0+c	a = b = c + 1	
1/15/2010		MET CS 5635 2. Programming F	Spring 2010 Fundamentals		16



Logic	al operators	5		
OR:	operand1	operand2	operand1 operand2	
	true	true	true	
	true	false	true	
	false	true	true	
	false	false	false	binary
AND:	operand1	operand2	operand1 & & operand2	
	true	true	true	
	true	false	false	
	false	true	false	
	false	false	false	2
NOT:	operand	!operand)
	true	false		unary
	false	true		







	Output Strea	m (continued)
endl	vs.	''\n'' or '\n'
Both will start a ne	wline	
'\n' inserts a char in	n the output stream	nothing more
endl is a stream matchange the form the buffer and d	nipulator, i.e. it is natting. In addition isplays all its conto	a function used with >> and << to to inserting '\n\' it also empties or <i>flushes</i> nt. In contrast to this while when
	1. 1 1 1	
inserting '\n' the architecture). H flush every time screen. Differen	a display <i>may</i> be de lowever, most stread when seeing a '\n aces often occur wh	layed ("may" as it varies with the ms are line buffered by default, i.e. they —thus there will be no difference on the en there is combined use of << and >>
inserting '\n' the architecture). H flush every time screen. Differen It is safer to use en	a display may be de lowever, most stread when seeing a '\n aces often occur when the occur when	layed ("may" as it varies with the ms are line buffered by default, i.e. they —thus there will be no difference on the en there is combined use of << and >> ure flushing.
Inserting '\n' the architecture). H flush every time screen. Differen It is safer to use en	Isplay may be de lowever, most stree when seeing a '\n nees often occur wh dl with cout to ens	layed ("may" as it varies with the ms are line buffered by default, i.e. they —thus there will be no difference on the en there is combined use of << and >> ure flushing.



Casts The <i>static cast</i> op conversion of one	perator is an unary operator that e data type into another	t allows explicit
Syntax: ke	yword Angle brackets are type is the new data	part of syntax and a type
	static_cast< type > (express	sion)
e.g.: static_cast	<double> (totalStudents)</double>	expression whose type is changed
precedence high	er than unary + and -, same as j	postincrement
<u>associativity</u> ri (See Appendix C: Oper	ght to left (unary pre-fix operat ator Summary [Ho 09])	tor)
The remainin reinterpret_o	g cast operators dynamic_c ast, const_cast will be dis	ast, ocussed later
1/15/2010	MET CS 563Spring 2010 2. Programming Fundamentals	5

C-like notation	
(type) identifier	e.g.: (double) totalStudents
Functional notation	
type (identifier)	e.g.: double(totalStudents)
• Old style casts continue to when applied to classes a	b be available but <i>can easily lead to errors</i> nd thus <i>should be used only if new styles</i>
<i>are not available</i> . For mo	ore detail on differences between old and
new style casts see	/* // · · · · · · · · ·

int i=2, j=3; cout<<"Integer operations:\n"; cout<<"i = " << i << ", j = " << j << cout<<"(i+j)/2 = "<<(i+j)/2 < <endl; cout<<"double value of $(i+j)/2 =$ "< cout<<"double value of $(i+j)/2 =$ "< cout<<"label{eq:cout} operations:\n"; cout<<"x = " << x << ", y = " << y + cout<<"(x+y)/2 = "<<(x+y)/2 <<endle operations:\n";<br="">cout<<"(x+y)/2 = "<<(x+y)/2 <<=ndle operations:\n"; cout<<"(x+y)/2 = "<<(</endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endle></endl; 	<pre>< endl; << static_cast<double> ((i+j))/2 <<endl; << double ((i+j))/2 <<endl; idl; << endl; dl; <<endl; illegal="" operation<="" pre=""></endl;></endl; </endl; </double></pre>
Integer operations: i = 2, j = 3 (i+j)/2 = 2 double value of($(i+j)/2 = 2.5$ double value of($(i+j)/2 = 2.5$ (i+j)%2 = 1	Double operations: x = 2, y = 3 (x+y)/2 = 2.5
1/15/2010 MET C 2. Progra	26 amming Fundamentals



2	2.9. Ranges of Fundamental Types.	
The range of nur number of bytes	nbers represented by a given data type is det allotted to this type in memory.	ermined by the
This size is mach	nine dependent and can be checked by the size	zeof() operator.
Independent of the valid for all C++	he size for a specific architecture the followi implementations:	ng restrictions are
$1 \equiv sizeof(c)$	$har) \le sizeof(int) \le sizeof(float) \le sizeof($	louble)
The range of son may modify the Modify the Siz	ne basic data types can be modified through	the <i>qualifiers</i> that
• Moaijy ine Siz	e. short, long	
• Use Sign bit di	ifferently while keeping the same byte size: u	nsigned, signed
1/15/2010	MET CS 563Spring 2010 2. Programming Fundamentals	28



Туре	В	Other Name	Range	
char	1		[-128, 127]	
unsigned char	1		[0, 255]	
short	2	short int, signed short int	[-32,768; 32,767]	
unsigned short	2	unsigned short int	[0; 65,535]	
int	4	signed	[-2,147,483,648; 2,147,483,647]	
unsigned int	4	unsigned	[0; 4,294,967,295]	
long	4	long int, long signed int	[-2,147,483,648; 2,147,483,647]	
unsigned long	4	unsigned long int	[0; 4,294,967,295]	
long long	8		[-9,223,372,036,854,775,808; 9,223,372,036,854,775,807]	
float	4		3.4E +/- 38 (7 significant digits)	
double	8		1.7E +/- 308 (15 digits)	
long double	8		1.7E +/- 308 (15 digits)	
More details on ranges see To visualize byte values, see powers of 2 chart <u>http://www.vaughns-1-</u>				
1/15/2010 MET CS 563Spring 2010 30 2. Programming Fundamentals				



cout << "Size of cl	har is " << sizeof(char)< <endl;< td=""><td></td></endl;<>			
cout<<"Size of bool is " << sizeof(bool)< <endl;< td=""></endl;<>				
cout<<"Size of sl	nort int is " << sizeof(short)< <endl;< td=""><td></td></endl;<>			
cout << "Size of in	nt is " << sizeof(int)< <endl;< td=""><td></td></endl;<>			
cout << "Size of lo	ong int is " << sizeof(long)< <endl;< td=""><td></td></endl;<>			
cout << "Size of float is " << size of (float) << endl;				
cout << "Size of de	ouble is " << sizeof(double)< <endl;< td=""><td></td></endl;<>			
cout<<"Size of lo	ong double is " << sizeof(long double)< <end< td=""><td>dl;</td></end<>	dl;		
		,		
	Size of char is 1			
	Size of bool is 1			
	Size of short int is 2			
	Size of int is 4			
	Size of long int is 4			
	Size of float is 4			
	Size of double is 8			
	Size of long double is 8			
1/15/2010	MET CS 563Spring 2010	31		
	2. Programming Fundamentals			



defines the <i>temp</i> and maximum s floating point no	<i>plate class</i> numeric_limits to access accuracy, minimizes, rounding, and signaling type errors for integral sumbers	um and
Usage:		
numeric_	limits <type>:: function</type>	
e.g.		
numeric_	limits <int>:: max()</int>	
numeric_	limits <int>:: min()</int>	
numeric_	limits <double>:: max()</double>	
numeric_	limits <double>::round_error()</double>	
For other functi	ons see numeric limits members at	
http://www.cplu	<u>usplus.com/reference/std/limits/numeric_limits/</u> or	
http://msdn.mic	rosoft.com/en-us/library/ct1as7hw.aspx and more	
examples at <u>htt</u>	<u>p://msdn.microsoft.com/en-</u>	
us/library/85084	4kd6(VS.80).aspx	
/15/2010	MET CS 563Spring 2010 2. Programming Fundamentals	

<cfloat> and <climits> (previously float.h and limits.h) define names for the system dependent characteristics of floating and integral types respectively:</climits></cfloat>		
cout<<"Smallest int << INT_MIN;	value is INT_MIN = "	
FLT_MIN, FLT_MIN	float minimum, maximum	
DBL_MIN, DBL_MAX	double minium, maximum	
LDBL_MIN, LDBL_MAX	long double minimum, maximum	



char overflow (sour cout<<"Character Overfl char c = CHAR_MAX - 1 cout<<"char c = Char_M cout<<"c+1 = " <c+1<< cout<<"c+2 = "<c+2<< cout<<"c+3 = "<c+3<< cout<<"c+4 = "<c+4<< p=""></c+4<<></c+3<< </c+2<< </c+1<< 	rce file L2-6-intOverflow.c $ow \n";$ 2; AX - 2 = "< <c<endl; " cast to char " << char(c+1)<<endl " cast to char " << char(c+2)<<endl " cast to char " << char(c+3)<<endl " cast to char " << char(c+3)<<endl " cast to char " << char(c+4)<<endl< th=""><th>; ;; ;; ;;</th></endl<></endl </endl </endl </endl </c<endl; 	; ;; ;; ;;
cout<<"Character Under c = CHAR_MIN + 2; cout<<"char c = CHAR_ cout<<"c-1 = "< <c-1<" cout<<"c-2 = "<<c-2<" cout<<"c-3 = "<<c-3<" cout<<"c-4 = "<<c-4<"< td=""><td>flow \n"; MIN + 2 = "<<c<<endl; cast to char " << char(c-1)<<endl; cast to char " << char(c-2)<<endl; cast to char " << char(c-3)<<endl; cast to char " << char(c-4)<<endl;< td=""><td>Character Overflow char c = Char_MAX - 2 = } c+1 = 126 cast to char \sim c+2 = 127 cast to char \triangle c+3 = 128 cast to char \bigcirc c+4 = 129 cast to char \bigcirc Character Underflow char c = CHAR_MIN + 2 = é c-1 = -127 cast to char \bigcirc c-2 = -128 cast to char \bigcirc c-3 = -129 cast to char \triangle c-4 = -130 cast to char \sim</td></endl;<></endl; </endl; </endl; </c<<endl; </td></c-4<"<></c-3<" </c-2<" </c-1<" 	flow \n"; MIN + 2 = "< <c<<endl; cast to char " << char(c-1)<<endl; cast to char " << char(c-2)<<endl; cast to char " << char(c-3)<<endl; cast to char " << char(c-4)<<endl;< td=""><td>Character Overflow char c = Char_MAX - 2 = } c+1 = 126 cast to char \sim c+2 = 127 cast to char \triangle c+3 = 128 cast to char \bigcirc c+4 = 129 cast to char \bigcirc Character Underflow char c = CHAR_MIN + 2 = é c-1 = -127 cast to char \bigcirc c-2 = -128 cast to char \bigcirc c-3 = -129 cast to char \triangle c-4 = -130 cast to char \sim</td></endl;<></endl; </endl; </endl; </c<<endl; 	Character Overflow char c = Char_MAX - 2 = } c+1 = 126 cast to char \sim c+2 = 127 cast to char \triangle c+3 = 128 cast to char \bigcirc c+4 = 129 cast to char \bigcirc Character Underflow char c = CHAR_MIN + 2 = é c-1 = -127 cast to char \bigcirc c-2 = -128 cast to char \bigcirc c-3 = -129 cast to char \triangle c-4 = -130 cast to char \sim
1/15/2010	MET CS 563Spring 2010 2. Programming Fundamentals	36

\triangleright	literals:				
	decimal (radix 10)	2	63	83	
	octal (radix 8) prefix zero(0)	02	077	0123	
	hexadecimal (radix 16) prefix zero-x(0x)	0x2	0x3f	0x53	
	suffixes: U for unsigned, L for long: 1234	156789	L		
\triangleright	range depends on whether signed or unsi	gned			
	e.g. for short int of 2B :				
	unsigned: $65536=2^{16}$ distinct values $\in [0, \infty)$, 65535	5]		
	signed: 32768=2 ¹⁵ negative and 32768 no	on-nega	ative(0 ar	nd 32767	
	positive) values				
	∈[-32768, +32767]				
	Warning: not a good idea to use unsigned	l to gai	n 1 more	bit: it is	
	obliterated by promotion rules any way.				
	Tip: Because of				
	$1 \equiv sizeof(char) \leq sizeof(short) \leq sizeof(i$	$nts) \le s$	izeof(lor	ng)	
	integers are either 16 or 32 b depending of	on the h	ardware		
	but short are 2B and long are 4B on pra	ctically	every m	achine	
	overflow no error message given!!!				
> 1/1	overflow no error message given!!! 5/2010 MET CS 563Spring 2	010			37

int range (source file: L2-5-typeLimits.cpp)			
<pre>cout<<"\nLargest short int value is numeric_limits<short int="">::max() = "</short></pre>			
<< numeric_limits <short int="">::max() << endl;</short>			
<pre>cout<<"Smallest short int value is numeric_limits<short int="">::min() = "</short></pre>			
<< numeric_limits <short int="">::min() <<endl;< td=""></endl;<></short>			
cout << "\nLargest int value is INT_MAX = " << INT_MAX < <endl;< td=""></endl;<>			
cout<<"Largest int value is numeric_limits <int>::max() = "</int>			
<< numeric_limits <int>::max() << endl;</int>			
cout<<"Smallest int value is INT_MIN = " << INT_MIN < <endl;< td=""></endl;<>			
<pre>cout<<"Smallest int value is numeric_limits<int>::min() = "</int></pre>			
<< numeric_limits <int>::min() <<endl;< td=""></endl;<></int>			
<pre>cout<<"\nLargest long int value is numeric_limits<long int="">::max() = "</long></pre>			
<< numeric_limits <long int="">::max() << endl;</long>			
<pre>cout<<"Smallest long int value is numeric_limits<long int="">::min() = "</long></pre>			
<< numeric_limits <long int="">::min() <<endl;< th=""></endl;<></long>			
Largest short int value is numeric_limits <short int="">::max() = 32767</short>			
Smallest short int value is numeric_limits <short int="">::min() = -32768</short>			
Largest int value is INT_MAX = 2147483647			
Largest int value is numeric_limits <int>::max() = 2147483647</int>			
Smallest int value is INT_MIN = -2147483648			
Smallest int value is numeric_limits <int>::min() = -2147483648</int>			
Largest long int value is numeric_limits <long int="">::max() = 2147483647</long>			
Smallest long int value is numeric_limits <long int="">::min() = -2147483648</long>			
2. Programming Fundamentals			



cout<<"Unsigned unsigned int U_M unsigned int nonN cout<<"unsigned	Integer Underflow \n"; IN = numeric_limits <unsigned int="">::min(); fegative = U_MIN + 2; int nonNegative = U_MIN + 2 = "<<nonnegative<<endl;< th=""><th></th></nonnegative<<endl;<></unsigned>	
cout << "nonNegat	ive-1 = "< <nonnegative-1<<endl;< td=""><td></td></nonnegative-1<<endl;<>	
cout<<"nonNegative-2 = "< <nonnegative-2<<endl;< td=""></nonnegative-2<<endl;<>		
cout<<"nonNegative-3 = "< <nonnegative-3<<endl;< td=""></nonnegative-3<<endl;<>		
	Unsigned Integer Underflow unsigned int nonNegative = U_MIN + 2 = 2 nonNegative-1 = 1 nonNegative-2 = 0 nonNegative-3 = 4294967295 nonNegative-4 = 4294967294	
1/15/2010	MET CS 563Spring 2010	40



Floating Point	range (source file: L2-5-typeL	Limits.cpp)
cout<<"\nLarges << numeric cout<<"Smallest << numeric cout<<"\nLarges << numeric cout<<"Smallest << numeric cout<<"\nLarges << numeric cout<<"\nLarges << numeric	t float value is numeric_limits <float>::n _limits<float>::max() << endl; float value is numeric_limits<float>::m _limits<float>::min() << endl; t double value is numeric_limits<doubl _limits<double>::max() << endl; double value is numeric_limits<double _limits<double>::min() << endl; t long double value is numeric_limits<1 _limits<long double="">::max() << endl;</long></double></double </double></doubl </float></float></float></float>	<pre>max() = " nin() = " le>::max() = " e>::min() = " long double>::max() = "</pre>
cout<<"Smallest << numeric	long double value is numeric_limits <lo _limits<long double="">::min() <<endl;< td=""><td>ong double>::min() = "</td></endl;<></long></lo 	ong double>::min() = "
Largest float value Smallest float value	e is numeric_limits <float>::max() = 3.4 ue is numeric_limits<float>::min() = 1.</float></float>	40282e+038 17549e-038
Largest double va Smallest double v	lue is numeric_limits <double>::max() = ralue is numeric_limits<double>::min()</double></double>	= 1.79769e+308 = 2.22507e-308
Largest long doub Smallest long dou	ble value is numeric_limits <long double<br="">ble value is numeric_limits<long double<="" td=""><td>e>::max() = 1.79769e+308 le>::min() = 2.22507e-308</td></long></long>	e>::max() = 1.79769e+308 le>::min() = 2.22507e-308
1/15/2010	MET CS 563Spring 2010 2. Programming Fundamentals	42



Unexpected but	t Common Error Sources		
 Entering a float of program failure, of values for pennie the remaining entering 	or a character when an integer value is required e.g. program L2-7-intInput.cpp asks the user is, nickels, dimes, and quarters; if a char or floa tries are considered 0 and the program terminat	results in to input int at is entered tes.	
- Typing in the value of the ventually lead to be in programing the cans; this has the in one single place of the cans is the single place of the canal sector of the	ue of a constant, e.g. π , every time it is needed o an error. Instead it is better to define a constant m L2-8-constDefinition.cpp for the volume of additional benefit that if changes are needed the ce.	will nt object as f bottles and ney are done	
 precision is deter sometimes 19) ar whenever the ma is well within the some examples. 	mined by mantissa length, (for double typically nd is <u>independent</u> from range of the number; The ntissa length is exceeded an error occurs even by type range; The program in L2-9-precision.cp	y 15, nus if the value pp shows	
 Some floating point numbers, e.g. 4. 35, do not have an exact binary representation (similarly 1/3 does not have an exact decimal representation_, and round-off errors can produce very strange results(see L2-10-roundOffError.cpp for examples). 			
1/15/2010	MET CS 563Spring 2010 2. Programming Fundamentals	44	



The preproce and replaces Standard dire	ssor takes entire file specified as the directive argument <i>fileName</i> he directive with the text of the file. etory file names are enclosed in angle brackets <>:
# include <io< td=""><td>stream></td></io<>	stream>
# include <cn< td=""><td>hath></td></cn<>	hath>
User defined	files names are enclosed in double quotes " " :
# include "c:\	cs563\examples\myHeaderFile.h"

using namespace Directive <i>identifies</i> th group (scope) to which the header files belong (not a preprocessing directive)	e # include <iostream> using std :: cout using std :: cin</iostream>
# include <iostream></iostream>	using std :: endl
int main(){	int main(){
std :: cout << " Enter integer: ".	cout << " Enter integer: ":
std :: cin >> count:	cin >> count:
std :: cout << " You entered " << count << en	ndl: cout << " You entered " << count << endl;
}	}
<pre># include <iostream> using namespace std ; ends with ;</iostream></pre>	# include <iostream.h> // old style</iostream.h>
int main(){	int main(){
<pre> cout << " Enter integer: "; cin >> count;</pre>	 cout << " Enter integer: "; cin >> count;
cout << " You entered " << count << endl;	cout << "You entered "<< count << endi;
	}
j) 55Spring 2010 4/
2. Programm	ing Fundamentals

