Function Name	**Bug Name**	**Explanation of the Bug**	**Example Bad Code that will trigger it**	**What it should print to the screen**	Date Completed (Stable || Naive)	Notes	Bugs
bug1	Cout backwards arrows	Student uses the wrong stream operator with a cout statement	cout >> x;	Hey, it looks like you're trying to do a cout statement. In order to do a cout statement, you want to type something like this: cout << x; Note that it looks like there are arrows pointing FROM the variable TO cout. I think you might have done yours backwards (cout >> x; is wrong).	June 27 - stable - WRK		
bug2	Cin backwards arrows	""	""	""	June 27 - stable - WRK		
bug3	Missing semicolon	Student forgets to end a line of code with a semicolon, which is required in C++.	cout << x	You've ended a line of code without a semicolon at <Line Number>. A semicolon is required for non-conditional lines of code.	July 12th, 2016 -> 3:42 AM Naive - TLL Comments: this is very stable running, won't cause a crash. However, it is very naive. Need serious flexibility tuning.		Fails on if statements, else statements, and some other stuff. Check out /tmp/bumain.cc
bug4	Unitialized variable	Student does not follow declaration of basic types (for this class and for C++ISO Standard).	int x;	You've declared a variable of a basic type without initializing a value for it. This is not a strict compiler error, but may cause undefined behavior if the integer is used as a loop control variable without previous modification of it's value. The integer declared could hold any value between negative two billion and positive two billion if not initialized.	July 1st, 2016 -> 7:56 PM Stable - TLL		Occasional crashing Should be fixed - TLL
bug5	Missing semicolon after a class	Student doesn't follow class declaration standard.	class Foo {}	You've declared a class without an ending semicolon. This semicolon is required in C++.	June 30th, 2016 -> 7:56 Stable - KR		

bug6	Extra semicolon after a for loop	Student mistakes for declaration as a normal line of code.	for (int i = 0; i < 10; i++);	A for loop does not require a semicolon after it's declaration. A proper for loop is constructed as follows.
				for (int i = 0; i < 10; i++) {
				    code;
				    code;
				    code;
				}
					June 30th, 2016 -> 3:50 PM
					Stable - TLL
						Original version was by KR, which got wiped out before we started using version control.	

bug7	Improper for loop	Doesn't follow the code pattern I teach in class	for (int i = 1; i <= 10; i++)	In general, you should try to have your for loops start with 0 and end with a < X, where X is the number of times you want the loop to run. This is not a hard and fast rule, though, so if you know what you're doing, go ahead and hit "override" here.	June 30th, 2016 -> 3:50 PM Stable - TLL		Occasional crashing Should be fixed - TLL
bug8	Dereferencing a null pointer	Student doesn't keep track of null state of an established pointer.	int *x = null_ptr; cout << *x;	You've dereferenced x. Which is an integer pointer currently assigned as NULL. This means it's pointing at the memory address 0x0. This is a non-writable zone of memory. Make sure when dereferencing a pointer it will have a valid memory address. https://www.youtube.com/watch?v=bLHL75H_VEM		Possibly unneeded: (Reason: CSCI 41 only)	
bug9	Single equals in if statement	Student is confusing assignment vs comparison.	if (x = 42)	You've tried to assign the integer x the value 42 while in a conditional if or else if statement. If you wanted to compare integer x to the value 42, you must use two equal signs (==).	June 30th, 2016 -> 5:50 PM Stable - WRK	This bug also checks for not having a closing parenthesis and an if statement that begins or ends in an equal sign, both of which are bad and give errors	
bug10	Single | and & in if statements	Student isn't using proper logical operator syntax for || and &&.	if (x | y)	You've not used the || or && operator properly. A single | or & is not a valid logic operator. (This needs to be reworded, I'm tired -Taylor)	June 30th, 2016 -> 3:50 PM Stable - TLL		
bug11	Caret operator	Student using caret operator as the "power" function, in C++ this symbol is used for XOR.	cout << 5 ^ 3;	You've used the caret operator (^) with two integers, probably expecting it to raise one integer to the other. This operator in C++ is responsible for XOR (exclusive OR). If you want to raise an integer, use the <cmath> function pow. Declaration of pow -> pow(x,y). Where x and y are of types, int or double, concurrently.	June 30th, 2016 -> 8:24 PM Stable - KR		
bug12	Bad indentation	Student isn't following coding best practices for style, which increases readibility.	Lines in code blocks on the first column	Make sure to indent your code after opening a pair of braces ({}). This increases readibility for people helping you with your code. It also increases your ability to track down bugs and review your code before refactoring.		Stylyzing? Possibly just teach good stylizing? Also, gg=G	
bug13	Incorrectly variable sized array	Student isn't correctly establishing a variable sized array.	int arr[x]	To declare a variable sized array. You must invoke the new function. As so, int *p = new arr[x]; This will create an integer pointer to the beginning of an int arr with size x.	July 2nd, 2016 -> 7:49AM Stable - KR		
~~bug 14~~	~~Unused variables~~	~~A variable has been declared but is not used.~~	~~int x = 0; and x does not occur again in file.~~	~~Unused variable x on line <line no>. The compiler will optimize this variable out, but it's nice to not have extra lines to read.~~	~~Naive - KR~~	~~Ignore? -Wall should pick this up. (Or -Wunused-variable)~~	
~~bug 15~~	~~Using undeclared variables~~	~~A variable has not been declared but is used.~~	~~int main() {~~ ~~cin >> x;~~ ~~cout << x;~~ }	~~variable x has not been declared, you cannot use a variable without first declaring it.~~		~~Error message is short. Might not include support for this.~~ ~~-JN~~ ~~Exact error message for example error:~~ ~~main.cc:8:6: error: ‘x’ was not declared in this scope~~ ~~cin>>x;~~	

bug 16	Undeclared functions	A function is being called without the proper prototype or defintion antecedent to the call.	int main() {
			    menu();
			}
			void menu() {
			    ....;
			}
						Exact error message for example error:
						main.cc:8:6: error: ‘menu’ was not declared in this scope
						 menu();	

bug 17	Variable scoping issue	A variable is being used out of scope	int main() {
			    int y = 1;
			    if(y == 1) {
			        int x = 5;
			        x++;
			    }
			    cout << x;
			}
				Variable x is out of scope. After a condition statement such as if,elseif,else, all variables declared within the condition will be destructed on the exit bracket.		Error message is short. Might not include support for this.
						-JN
						Exact error message for example error:
						main.cc:14:7: error: ‘x’ was not declared in this scope
						 cout<<x;	

bug 18	Incorrect pointer/function operator	Using the wrong access operator when working with pointers	class Node {
			public:
			int value;
			*Node next;
			Node(){...}
			get_next(){...}
			get_value(){...}
			};
			
			int main(){
			Node* my_node;
			Node* some_node;
			Node* foo;
			my_node->next = some_node;
			some_node.next=foo; //bad
			return 0;
			}
				Did you know that pointers don't use the dot operator when they are accessed? Trying to do so can cause some pretty bad issues and lots of confusion. Use the -> operator instead.		Pointers not used in CSCI 40. Possibly remove support.
						Also pointers are a good Cs learning topic, and trial by fire.
						JN	

bug 19	Infinite for loops	Making a for loop that will run infinitely	for(i=100; i>100; i++){ ...}
			for(i=0; i<100, i--){...}
				Infinite for loop detected at line X. Did you make sure that you were iterating the right direction, and that your sign was facing the right way?	July 15th, 2016 -> 4:01 AM
					Stable - TLL
					Comments: This only checks for loops that follow the style and structure taught in the class. It is a naive check. Will not cause any crashes.		

bug 20	Improper Syntax	Improper syntax for scanf *Note: (no & for strings)	int num = 3;
			scanf("%d", num);
				Scanf requires you to attach an & right before the second parameter.
				Ex. scanf("%d", &num);
				However, you do not need to add the & when the variable of type string.
					June 30th, 2016 -> 9:36 PM
					Stable - KR		

bug 21	Improper Syntax	Improper sytanx for a for loop.	for (int i = 0, i < 10, i++)	A for loop's parameters are separated by a semicolo ';'
				Ex.
				for(int i = 0; i < 10; i++) {
				    cout << i << endl;
				}
					July 1st, 2016 -> 6:45 PM
					Stable - TLL
					
					Comments: This only check for common charcter mistakes in a for loop. This now also checks range-based for loops for correct syntax.
					
					E.g.: colon, comma		

bug 22	Improper array initialization	Array isn't being initialized correctly.	Int main(){
			int arr1 [5]={}; //good
			int arr2(5)={}; //bad
			return 0;
			}
				Arrays are initialized using the square brackets that look like [] not with parentheses ();	July 7th, 2016 -> 3:25 PM
					Stable - JAN
					
					Comment: Checks for basic array initialization using a check to see if there is an "(" on the same line as an int/double/float/char. Ignores lines that contain "vector","for","while"and "main" as the likelihood of an array being initialized on lines containing these key words is minimal.		

bug 23	Loss of data	User isconverting data types that might result in loss of data	int main(){
			int my_int=10;
			double my_double =4.20;
			my_int=my_double;
			}
				Doubles are not like ints in c++. Ints can only hold whole numbers and any decimal that they would have is actually cut off so the number is rounded down. If you wanted to express the whole value, make sure you store values into those of the same type. (Doubles with doubles, ints with ints, etc).	July 8th, 2016 -> 5:10 AM
					Stable - TLL		

bug 24	Incorrect code command	Using a Visual Studios OS command in Unix	int main(){
			cout<<"Hello World"<<endl;
			system("PAUSE");
			return 0;
			}
				System("Pause") actually isn't propper coding convention. It's something that is native only to Windows and is not available on Unix which is the type of server you are currently using. Please make sure to remove this from your code.	July 1st, 2016 -> 5:00 PM
					Stable - TLL		

bug 25	Incorrect variable naming convention	Using a numeric or punctuation at the beginning when naming variables	int main(){
			int 1st_number=5;
			int 2nd_number=10;
			string .com_name="my_website.com";
			
			return 0;
			}
				Proper coding convention states that you do not start any of your variable names with a numeric or a punctuation mark. Please make sure that none of your variables do this.	July 1st, 2016 -> 4:39 PM
					Stable - TLL		

bug 26	Improper Syntax	Using one too many or too few parentheses	int num1=5;
			int num2=3;
			if(num1)+(num2+2)==10)
				Making sure that you have the appropriate amount of parenthesese is crucial to making sure that your code runs properly.
				You should count them again on line X to make sure that they match exactly as you want.
					July 1st, 2016 -> 7:14 PM
					Stable - TLL		

bug 27	Improper Syntax	Forgetting the # before #include	include<iostream>	You are trying to include a c++ library file but you forgot the # before the #include.	June 30th, 2016 -> 5:22 PM Stable - TLL		
bug 28	Lack of Magic Use	Forgetting using namespace std &| lack of including <iostream>		You forgot to add #include <iostream> and or didn't clarify your namespace to be using namespace std. This is not a strict compiler error and you can override if you're sure your code is correct.	July 9th, 2016 -> 5:34 PM Stable - TLL		