Activity #5: Boolean Expressions
Recorder's Report

Manager: Reader:

Recorder: Driver:

Date: Score: Satisfactory / Not Satisfactory

Record your team's answers to the key questions (marked with ) below.

a) Model 1, Question #5

b) Model 2, Question #11

c) Model 3, Question #14

Activity #5: Boolean Expressions

In this course, you will work in teams of 3–4 students to learn new concepts. This activity will introduce you to boolean expressions and branching in C++.

Content Learning Objectives

After completing this activity, students should be able to:

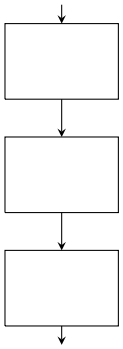
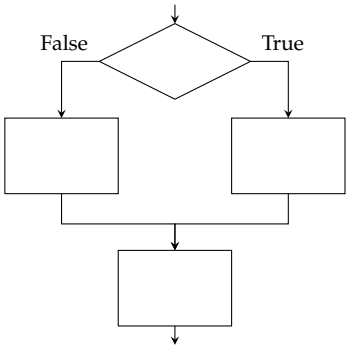
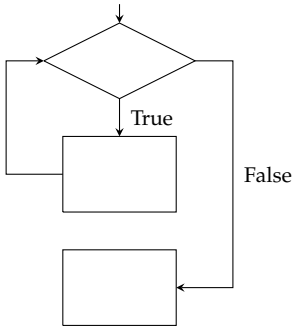
- Explain sequential, branching, and looping programming structures.
- Explain how relational and logical operators are used in programming.
- Use conditional operators with strings and numeric values.

Process Skill Goals

During the activity, students should make progress toward:

- Write correct Boolean expressions and compound expressions.

Model 1 Programming Structures

Sequential Structure	Branching Structure	Looping Structure
		

Refer to Model 1 above as your team develops consensus answers to the questions below.

Questions (20 min) **Start time:**

1. Based only on the pictures in the model above, what do the shapes represent?

a) Rectangle

b) Diamond

2. Which structure best describes the types of programs you've written thus far?

3. Which structure(s) allows the programmer to create programs that decide what code to execute?

4. A *relational operator* is used to test for a particular relationship between two values. It returns either **true**, if the relationship holds, or **false**, if it does not. What relationship does each operator below test for? If unsure, use the file `activity05a.cpp` to try them out.

a) $<$

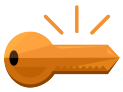
b) $<=$

c) $!=$

d) $>$

e) $>=$

f) $==$

5. Use the variable values below to determine the value of the following expressions. 

```
1 int x = 4, y = 5, z = 4;
2
```

a) `x > y`

b) `x < y`

c) `x == y`

d) `x != y`

e) `x >= z`

f) `x <= z`

g) `x + y > 2 * x`

h) `y * x - z != 4 % 4 + 15`

i) `pow(x,2) == abs(-16)`

6. What are the two possible answers for each expression in that last question?

7. Assume the following strings have been defined. Determine the results of the following expressions.

```
1 string word1 = "hello";
2 string word2 = "good-bye";
3
```

a) `word1 == word2`

b) `word1 != word2`

c) `word1 < word2`

d) `word1 >= word2`

8. How do relational operators work on strings?

Model 2 A Truth Table

Condition #1 `p`	Condition #2 `q`	Negation (NOT) `! p`	Conjunction (AND) `p && q`	Disjunction (OR) `p || q`
True	True	False	True	True
True	False	False	False	True
False	True	True	False	True
False	False	True	False	False

Refer to Model 2 above as your team develops consensus answers to the questions below.

Questions (15 min) Start time:

9. The symbols `&&`, `||`, and `!` are called *logical operators* because they combine conditions that are either true or false to create new compound conditions. Given the variable definitions below, fill in the appropriate logical operator to produce the desired truth value.

```
1  int numBooks = 40;
2
```

a) `(numBooks > 5)` `(numBooks < 100)` – this should be **true**.

b) `(numBooks < 5)` `(numBooks > 20)` – this should be **true**.

c) `(numBooks * 10 == 400)` – this should be **false**.

10. A *Boolean Expression* is an expression that uses relational operators and/or logical operators together with variables, literal values, and the *Boolean* values "true" and "false". Translate the following Boolean expressions into an English statement. The first one is done for you.

a) `(x == 2) && (y > 3)` The variable x equals two and the variable y is bigger than three.

b) `(x != 4) || (y <= 7)`

c) `(x >= 2) && (x <= 10)`

d) `!((x == 2) && (y == 1))`

11. Write a Boolean Expression for each English statement.

a) The string `name` is not equal to "Jane".

b) The value of x is twice that of y or y is less than ten.

c) The value of z is between 0 and 5 excluding endpoints.

d) It is not the case that w is between 0 and 5 including endpoints.

Model 3 A C++ Program

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6    // declare variables
7    int a = 5, b = 2, c = 0;
8    // print output
9    cout << "Line 9:  " << (a == b) << endl;
10   cout << "Line 10:  " << (a != b) << endl;
11   cout << "Line 11: " << (a = b) << endl;
12   cout << "Line 12: " << ( (a > c) || ((b / c) == 1) ) << endl;
13   cout << "Line 13: " << ( (a > c) && ((b / c) == 1) ) << endl;
14 }
15
```

Refer to Model 3 above as your team develops consensus answers to the questions below.

Questions (15 min) Start time:

12. This program can be found in `activity05b.cpp`. Run it and determine the output produced by each `cout` statement.

a) The `cout` on line 9:

b) The `cout` on line 10:

c) The `cout` on line 11:

d) The `cout` on line 12:

e) The `cout` on line 13:

13. Based on the output observed in lines 9 and 10, how does C++ represent true and false?

14. Based on the output from lines 9 and 11, how does C++ treat = and == differently? 

15. In the code above, line 12 prints a value, while line 13 produces an error. Why do you think that is?