

2/23/19

Chapter 2 Written Homework

Eric Bollar

Exercises 1-11 (pgs. 105-107)

1. Mark the following statements as true or false.
 - a. An identifier can be any sequence of digits and letters.
 - i. **False**
 - b. In C++, there is no difference between a reserved word and a predefined identifier.
 - i. **False**
 - c. A C++ identifier can start with a digit.
 - i. **True**
 - d. The operands of the modulus operator must be integers.
 - i. **True**
 - e. If `a = 4;` and `b = 3;`, then after the statement `a = b;` the value of `b` is still 3.
 - i. **True**
 - f. In the statement `cin >> y;` `y` can only be an `int` or a `double` variable.
 - i. **False**
 - g. In an output statement, the newline character may be a part of the string.
 - i. **True**
 - h. The following is a legal C++ program:

```
int main() {  
    return 0;  
}
```

 - i. **True**
 - i. In a mixed expression, all the operands are converted to floating-point numbers.
 - i. **False**
 - j. Suppose `x = 5.` After the statement `y = x++;` executes, `y` is 5 and `x` is 6.
 - i. **True**
 - k. Suppose `a = 5.` After the statement `++a;` executes, the value of `a` is still 5 because the value of the expression is not saved in another variable.
 - i. **False**
2. Which of the following are valid C++ identifiers?
 - a. **firstCPPProject**
 - b. **ProgrammingLecture2**
 - c. Mike'sFirstAttempt
 - d. **New_Student**
 - e. **POP_QUIZ**
 - f. **3feetIn1Yard**
 - g. C++Program2
 - h. **Quiz7**
 - i. Update Grade

- j. 4th
3. Which of the following is a reserved word in C++?
- a. **Const**
 - b. **include**
 - c. **Char**
 - d. **void**
 - e. **int**
 - f. **Return**
4. What is the difference between a keyword and a user-defined identifier?
- a. **Keywords are reserved words and predefined identifiers already built into the C++ language that cannot be redefined. A user-defined identifier is any variable that the user had named and created themselves.**
5. Are the identifiers firstName and FirstName the same?
- a. **No, identifiers are case-sensitive.**
6. Evaluate the following expressions:
- a. $36 / 5$
 - i. **7**
 - b. $18 - 32 / 6 * 3$
 - i. **3**
 - c. $80 \% 11$
 - i. **3**
 - d. $6 - 8 \% 11$
 - i. **-2**
 - e. $22.0 / 5$
 - i. **4.4**
 - f. $27 - 12 / 8.0$
 - i. **25.5**
 - g. $25 - 7 \% 3 + 8 / 3$
 - i. **26**
 - h. $18.0 + 5.0 * 3.0 / 4.0$
 - i. **21.75**
7. If $x = 5$, $y = 6$, $z = 4$, and $w = 3.5$, evaluate each of the following statements, if possible. If it is not possible, state the reason.
- a. $(x + z) \% y$
 - i. **3**
 - b. $(x + y) \% w$
 - i. **0.5**
 - c. $(y + w) \% x$
 - i. **4.5**
 - d. $(x + y) * w$
 - i. **38.5**
 - e. $(x \% y) \% z$
 - i. **1**

- f. $(y \% z) \% x$
i. **0**
- g. $(x * z) \% y$
i. **2**
- h. $((x * y) * w) * z$
i. **420**

8. Given:

```
int num1, num2, newNum;
double x, y;
```

Which of the following assignments are valid? If an assignment is not valid, state why. When not given, assume that each variable is declared.

- a. `num1 = 35;`
i. **Valid**
 - b. `newNum = num1 - num2;`
i. **Valid**
 - c. `num1 = 5; num2 = 2 + num1; num1 = num2 / 3;`
i. **Valid**
 - d. `num1 * num2 = newNum;`
i. **Valid**
 - e. `x = 12 * num1 - 15.3;`
i. **Valid**
 - f. `num1 * 2 = newNum + num2;`
i. **nonValid, cannot have operands on left side unless using '==' as a check and not an assignment.**
 - g. `x / y = x * y;`
i. **nonValid, cannot have operands on left side unless using '==' as a check and not an assignment.**
 - h. `num2 = num1 % 2.0;`
i. **nonValid, having 2.0 makes right side a decimal, not an integer. num2 is of type integer.**
 - i. `newNum = static_cast<int>(x) % 5;`
i. **Valid**
 - j. `x = x + y - 5;`
i. **Valid**
 - k. `newNum = num1 + static_cast<int>(4.6 / 2);`
i. **Valid**
9. Do a walk-through to find the value assigned to e. Assume that all variables are properly declared.

```
a = 3;
b = 4;
```

$c = (a \% b) * 6$; **c = 18**

$d = c / b$; **d = 4**

$e = (a + b + c + d) / 4$;

e is equal to 7.

10. Which of the following variable declarations are correct? If a variable declaration is not correct, give the reason(s) and provide the correct variable declaration.

a. $55 = \text{age}$;

i. **Incorrect, cannot assign age to the number 55. The correct statement should be: $\text{age} = 55$;**

b. $\text{char letter} = ' '$;

i. **Correct**

c. $\text{string message} = \text{'First test is on Monday'}$

i. **Incorrect, requires a ';' at the end of the statement.**

d. $\text{int one} = 5$;

i. **Correct**

e. int prime ;

i. **Correct**

f. $\text{double } x, y, z$;

i. **Correct**

11. Which of the following are valid C++ assignment statements? Assume that i, x, and percent are double variables.

a. $i = i + 5$;

i. **Valid**

b. $x + 2 = x$;

i. **Invalid, cannot assign $x + 2$ to x, must be other way around.**

c. $x = 2.5 * x$;

i. **Valid**

d. $\text{percent} = 10\%$;

i. **Invalid, cannot use percent sign this way (it is modulus), must instead use 0.1 (decimal form of percentage).**