Students:
Section 2.2 is a part of 2 assignments: CSC108 CH02.1-2.10 C2A
This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See this article for more info.

Includes: CA
Due: 02/04/2025, 11:59 PM EST

2.2 Variables (int)

Variable declarations

A ***variable declaration*** is a statement that declares a new variable, specifying the variable's name and type. Ex: `int userAge;` declares a new variable named userAge that can hold an integer value. The compiler allocates a memory location for userAge capable of storing an integer. ***Allocation*** is the process of determining a suitable memory location to store data like variables. Ex: In the animation below, the compiler has given userAge memory location 97, which is known as the variable's address. The choice of 97 is arbitrary and irrelevant to the programmer, but the idea that a variable corresponds to a memory location is important to understand.

When a statement that assigns a variable with a value executes, the processor writes the value into the variable's memory location. Likewise, reading a variable's value reads the value from the variable's memory location. The programmer must declare a variable before any statement that assigns or reads the variable, so that the variable's memory location is known.

PARTICIPATION ACTIVITY | 2.2.1: A variable refers to a memory location.

Start 2x speed

```
#include <iostream>
using namespace std;

int main() {
   int userAge;

   cout << "Enter your age: ";
   cin >> userAge;
   cout << userAge << " is a great age." << endl;

   return 0;
}
```

Memory
96
97 23 userAge
98
99

Enter your age: 23
23 is a great age.

Captions

1. Compiler allocates a memory location for userAge, in this case location 97.
2. First cout statement executes.
3. User types 23, cin assigns userAge with 23.
4. cout prints userAge's value to screen.

Feedback?

PARTICIPATION ACTIVITY | 2.2.2: Declaring integer variables.

Note: Capitalization matters, so MyNumber is not the same as myNumber.

1) Declare an integer variable named numPeople. (Do not initialize the variable.)

Check Show answer

2) Using two statements on two separate lines, declare integer variables named newSales and totalSales. (Do not initialize the variables.)

Check Show answer

3) What memory location (address) will a compiler allocate for the variable declaration below? If appropriate, type: Unknown

```
int numHouses = 99;
```

Check Show answer

Feedback?

Compiler optimization

Modern compilers may ignore unused variables, allocate variables on the stack, or use registers for variables. However, the conceptual view of a variable in memory helps understand many language aspects.

Assignment statements

An ***assignment statement*** assigns the variable on the left-side of the = with the current value of the right-side expression. Ex: `numApples = 8;` assigns numApples with the value of the right-side expression (in this case 8). [assign]

An ***expression*** may be a number like 80, a variable name like numApples, or a simple calculation like numApples + 1. Simple calculations can involve standard math operators like +, -, and *, and parentheses as in 2 * (numApples - 1). An integer like 80 appearing in an expression is known as an ***integer literal***.

In the code below, litterSize is assigned with 3, and yearlyLitters is assigned with 5. Later, annualMice is assigned with the value of litterSize * yearlyLitters (3 * 5, or 15), which is then printed. Next, litterSize is assigned with 14, yearlyLitters is assigned with 10, and annualMice is assigned with their product (14 * 10, or 140), which is printed.

Figure 2.2.1: Assigning a variable.

```
#include <iostream>
using namespace std;

int main() {
   int litterSize;
   int yearlyLitters;
   int annualMice;

   litterSize    = 3; // Low end of litter size range
   yearlyLitters = 5; // Low end of litters per year

   cout << "One female mouse may give birth to ";
   annualMice = litterSize * yearlyLitters;
   cout << annualMice << " mice," << endl;

   litterSize    = 14; // High end
   yearlyLitters = 10; // High end

   cout << "and up to ";
   annualMice = litterSize * yearlyLitters;
   cout << annualMice << " mice, in a year." << endl;

   return 0;
}
```

```
One female mouse may give birth to 15 mice,
and up to 140 mice, in a year.
```

Feedback?

PARTICIPATION ACTIVITY | 2.2.3: Assignment statements.

Be sure to end assignment statements with a semicolon (;).

1) Write an assignment statement to assign numCars with 99.

Check Show answer

2) Assign houseSize with 2300.

Check Show answer

3) Assign numFruit with the current value of numApples.

Check Show answer

4) The current value in houseRats is 200. What is in houseRats after executing the statement below? Valid answers: 0, 199, 200, or unknown.

```
numRodents = houseRats;
```

Check Show answer

5) Assign numItems with the result of ballCount - 3.

Check Show answer

6) dogCount is 5. What is in animalsTotal after executing the statement below?

```
animalsTotal = dogCount + 3;
```

Check Show answer

7) dogCount is 5. What is in dogCount after executing the statement below?

```
animalsTotal = dogCount + 3;
```

Check Show answer

8) What is in numBooks after both statements execute?

```
numBooks = 5;
numBooks = 3;
```

Check Show answer

Feedback?

CHALLENGE ACTIVITY | 2.2.1: Enter the output of the variable assignments.

620890.5010016.qx3zqy7

Start

Type the program's output

```
#include <iostream>
using namespace std;

int main() {
   int x;
   int y;

   x = 9;
   y = 6;

   cout << x << " " << y;

   return 0;
}
```

9 6

1 2 3 4 5 6

Check Next

Feedback?

Initializing variables

Although not required, an integer variable is often assigned an initial value when declared. Ex: `int maxScore = 100;` declares an int variable named maxScore with an initial value of 100.

Figure 2.2.2: Variable initialization: Example program.

```
#include <iostream>
using namespace std;

int main() {
   int avgLifespan = 70;
   int userAge;

   cout << "Enter your age: ";
   cin >> userAge;
   cout << userAge << " is a great age" << endl;

   cout << "Average lifespan is " << avgLifespan << endl;

   return 0;
}
```

```
Enter your age: 24
24 is a great age
Average lifespan is 70
```

Feedback?

PARTICIPATION ACTIVITY | 2.2.4: Declaring and initializing integer variables.

1) Declare an integer variable named numDogs, initializing the variable to 0 in the declaration.

Check Show answer

2) Declare an integer variable named daysCount, initializing the variable to 365 in the declaration.

Check Show answer

Feedback?

Assignment statement with same variable on both sides

Commonly, a variable appears on both the right and left side of the = operator. Ex: If numItems is 5, after `numItems = numItems + 1;` executes, numItems will be 6. The statement reads the value of numItems (5), adds 1, and assigns numItems with the result of 6, which replaces the value previously held in numItems.

PARTICIPATION ACTIVITY | 2.2.5: Variable assignments overwrite a variable's previous values: People-known example.

Start 2x speed

```
#include <iostream>
using namespace std;

int main() {
   int yourFriends;
   int totalFriends;

   cout << "Enter the number of people you know: ";
   cin >> yourFriends;
   totalFriends = yourFriends;
   cout << " You know " << totalFriends << " people.\n";
   totalFriends = totalFriends * yourFriends;
   cout << " Those people know " << totalFriends << " people.\n";
   totalFriends = totalFriends * yourFriends;
   cout << " And they know " << totalFriends << " people.\n\n";

   return 0;
}
```

Enter the number of people you know: 200
You know 200 people.
Those people know 40000 people.
And they know 8000000 people.

96 ??
97 200 yourFriends
98 8000000 totalFriends
99 ??

Captions

1. The compiler allocated memory for variables.
2. Prompt user with cout.
3. The cin statement assigns yourFriends with user input.
4. totalFriends is assigned with the value of yourFriends.
5. The cout statement outputs totalFriends.
6. The assignment statement reads totalFriends (200) and yourFriends (200), multiplies those values, and assigns totalFriends with the product of 40000.
7. The cout statement outputs totalFriends.
8. Assignment reads totalFriends (now 40000) and yourFriends (200), multiplies those values, and assigns totalFriends with the result of 8000000.

Feedback?

Six degrees of separation

The above example relates to the popular idea that any two people on earth are connected by just "six degrees of separation", accounting for overlapping of known-people.

PARTICIPATION ACTIVITY | 2.2.6: Assignment statements with same variable on both sides.

1) numApples is initially 5. What is numApples after:

```
numApples = numApples + 3;
```

Check Show answer

2) numApples is initially 5. What is numFruit after:

```
numFruit  = numApples;
numFruit  = numFruit + 1;
```

Check Show answer

3) Write a statement ending with - 1 that decreases the value of variable flyCount by 1.

Check Show answer

Feedback?

Common errors

A common error is to read a variable that has not yet been assigned a value. If a variable is declared but not initialized, the variable's memory location contains some unknown value, commonly but not always 0. A program with an uninitialized variable may thus run correctly on a system that has 0 in the memory location, but then fail on a different system—a very difficult bug to fix. A programmer must ensure that a program assigns a variable with a value before reading.

A common error by new programmers is to write an assignment statement in reverse. Ex: `numKids + numAdults = numPeople`*, or* `9 = beansCount`. Those statements won't compile, but writing `numCats = numDogs` in reverse *will* compile, leading to a hard-to-find bug.

PARTICIPATION ACTIVITY | 2.2.7: Common errors.

Which code segments have an error?

1) `21 = dogCount;`

- Error
- No error

2) `int amountOwed = -999;`

- Error
- No error

3)

```
int numDays;
int numYears;

numDays = numYears * 365;
```

- Error
- No error

Feedback?

CHALLENGE ACTIVITY | 2.2.2: Variables (int).

620890.5010016.qx3zqy7

Start

Declare an integer variable named numBreadfruits initialized with 55.

The output is: 55

Learn how our autograder works

```
#include <iostream>
using namespace std;

int main() {

   /* Your code goes here */

   cout << numBreadfruits << endl;

   return 0;
}
```

1 2 3 4

Check Next level

Feedback?

(*assign) We ask instructors to give us leeway to teach the idea of an "assignment statement," rather than the language's actual "assignment expression," whose use we condone primarily in a simple statement.

How was this section? Provide section feedback

Activity summary for assignment: CSC108 CH02.1-2.10 C2A

41 / 41 points

Due: 02/04/2025, 11:59 PM EST

41 / 41 points submitted to BlackboardLearn

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See this article for more info.

Completion details