

Can you figure them out?

L
O
V
E

TRY STAND
2

JOBINJOB

TAILR
RIALT
AIRTL
TLRIA

VARIABLES

ASSIGNMENT STATEMENTS

Luke Sathrum – CSCI 20

Today's Class

- Definition of a Variable
- Identifiers
- How to Declare Variables
- Variable Types
- Assignment Statements
- Uninitialized Variables
- Shorthand Assignment Statements

VARIABLES: IDENTIFIERS

Definition of a Variable

- A **variable** is the name for a piece of memory where we can store information
- Much like a
 - Mailbox
 - Cubbyhole
 - Box
- We can both put and retrieve information

Variable Examples

- This code would create variables that can hold whole numbers

```
int a, b, result;
```

- And this is some code to manipulate those variables

```
a = 5;
```

```
b = 2;
```

```
a = a + 1;
```

```
result = a - b;
```

Identifiers

- The name of a variable is called an **identifier**
- Must start with a letter or an underscore
 - C/C++ only
 - Other languages have their own rules
- The rest can be any combination of
 - Numbers
 - Letters
 - Underscores

Valid Identifiers in C/C++

- **x**
- **x1**
- **c_1_abc**
- **ABC123z7**
- **_sum**
- **RATE**
- **count**
- **data2**
- **bigBonus**

Invalid Identifiers in C/C++

- **12**
- **3X**
- **%change**
- **data-1**
- **myfirst.c**
- **PROG.CPP**

Identifiers are Case-Sensitive

- C++ Distinguishes between upper and lowercase letters
- The following are 3 different variables
 - `rate`
 - `RATE`
 - `Rate`
- We avoid using all 3
- Note that `cin`, `cout`, `main` are all lowercase

Identifiers: Naming Conventions

- Avoid starting with an underscore
- Start identifier with a lowercase letter
- Separate words with an underscore
- Make the name meaningful
- Good variables Names:
 - `top_speed`
 - `bank_rate1`
 - `bank_rate2`
 - `time_of_arrival`

Identifiers: Reserved Words

- Sometimes called keywords
- Have a predefined meaning
- Common ones
 - **int**
 - **long**
 - **float**
 - **double**
 - **char**

Identifiers: Reserved Words

- Note that `cin` and `cout` are NOT reserved words
- They are defined in `<iostream>`

Variables: Identifiers Summary

- Variables are where we store information in our programs
- The name of a variable is called an identifier
- There are special rules to how we name identifiers
- There are reserved words that cannot be used as identifiers

Sample Code

- Variables and Identifiers
 - `variables_identifiers.cpp`

VARIABLES: TYPES AND DECLARATIONS

Variable Types

- A variable needs to have a type associated with it
- Tells the variable what kind of information it can hold
- Three Main Types
 - Number
 - Integer
 - Floating-Point
 - Character
 - Boolean

Variable Types - Integer

Type Name / Keyword	Memory Used	Size Range
short (also called short int)	2 bytes	-32,768 to 32,767
int	4 bytes	-2,147,483,648 to 2,147,483,647
long (also called long int)	4 bytes	-2,147,483,648 to 2,147,483,647

Variable Types – Floating-Point

Type Name / Keyword	Memory Used	Size Range	Precision
float	4 bytes	approximately 10^{-38} to 10^{38}	7 Digits
double	8 bytes	approximately 10^{-308} to 10^{308}	15 Digits
long double	10 bytes	approximately 10^{-4932} to 10^{4932}	19 Digits

Variable Types – Character and Boolean

□ Character

Type Name / Keyword	Memory Used	Size Range
char	1 byte	All ASCII characters

□ Boolean

Type Name / Keyword	Memory Used	Size Range
bool	1 byte	Two keywords: true , false

Variable Types - String

- We have a type for a single character (**char**)
- What if we want a whole word
- We have a class called **string** that allows us to do this.
- Need to include this class

```
#include <string>
```

- It also is included automatically when we use **iostream**

```
#include <iostream>
```

- Use the keyword **string**

Variable Declaration

- You must declare before you use them
- Ex
 - `int number_of_beans;`
 - `double one_weight, total_weight;`
- Can declare anytime before using them

Variables Types and Declarations Summary

- Variables are where we store information in our programs
- Variables come in different types
 - Each type stores a certain kind of information
- Must be declared before we use them

Sample Code

- Declaring some Variables
 - `variables.cpp`

ASSIGNMENT STATEMENTS

Assignment Statements

- Change the value of a variable
- Use the assignment operator =
- Form:

variable = expression;

- Expression can be a combination of
 - Variables
 - Numbers
 - Operators
 - Function Invocations

Assignment Statements

- Evaluate right hand side of = and set the value of variable to the value of the expression
- Example (Assume our variables are declared)
 - `weight = 5;`
 - `total_weight = weight * 2;`
 - `count = 10;`
 - `count = count + 2;`

Assignment Statements

- An expression can be an assignment statement.

`n = (m = 2);`

- Same as

`n = m = 2;`

- What happens here?

- Sometimes we write:

`n = m = 2;`

- When we meant to write:

`n = m + 2;`

Variable Problems – Uninitialized Variables

- A variable has no meaningful value until we give it one
- We call these **uninitialized** variables
- These variables have garbage values
- Avoid by initializing variables when we declare

```
int minimum_number = 3;
```

```
double rate = 0.08, balance = 0.00;
```

- Also can write as

```
double rate(0.08), balance(0.00);
```

More Assignment Statements

- There are some shorthand assignment statements that combine an assignment operator and an arithmetic operator

`+=`

`-=`

`*=`

`/=`

`%=`

Shorthand Assignment Operators

Example	Equivalent To
<code>count += 2</code>	<code>count = count + 2</code>
<code>total -= discount</code>	<code>total = total - discount</code>
<code>bonus *= 2</code>	<code>bonus = bonus * 2</code>
<code>time /= rush_factor</code>	<code>time = time / rush_factor</code>
<code>change %= 100</code>	<code>change = change % 100</code>

Summary

- Assignment Statements change the value of a variable
- What is on the right side is stored on what is on the left side
- By default variables are uninitialized
- We have shorthand assignment statements

Sample Code

- Assigning values to variables and testing uninitialized variables
 - `assign.cpp`

Review

my_variable



6

Review

- There are rules to how we name identifiers
- There are some words we cannot use
- Variables need to have a type
- We use assignment statements to assign a value to a variable
- Variables start out with what value?