

---

## What Data Do We Have

---

## Today

- On Wednesday I showed you a C++ program that converts distances from miles to kilometers
- What are the main components of that program?
- Today we will
  - learn how C++ stores data
  - The different types of data that C++ can store

---

## Declaration Statements

- The declaration statements in the program are
  - `const double KM_PER_MILE = 1.609;`
  - `double miles;`
  - `double kms;`
- With the above statements I am declaring three things
  - `KM_PER_MILE` to store the conversion rate that never changes
  - `miles` to store the distance in miles as given by the user
  - `kms` to store the distance in kilometers as calculated by the program

---

## Declaration Statements

- Variable declarations
  - Allocate space for data to be used in the program
  - The data *can* be changed
  - `double miles;`
  - `double kms;`
- Constant declaration
  - Allocate space for data that *cannot* be changed
  - `const double KM_PER_MILE = 1.609;`

---

## Variable Declaration

- Variables are declared by stating
  - Type of data (data type)
  - Name to identify the variable (identifier)
  - Semicolon (;)
- `data-type identifier;`
- `double miles;`

---

## Variable Declaration

- If there is more than one variable of a single data type then you
  - State the data type
  - List the variable identifiers (names) separated by commas
  - Semicolon
- `data-type identifier1, identifier2;`
- `double miles, kms;`

## Data types and Identifiers

- Data types
  - C++ can store many different types of data
  - A data type also defines what operations can be performed on data of that type
  - We will start with the three primitive data types
    - `int` (whole numbers)
    - `double` (real numbers)
    - `char` (characters)
  - These data types must be in lower case
- Identifiers
  - Valid variable names in C++

09/03/04

CS150 Introduction to Computer Science 1

7

## `int`

- The `int` data type is used to store whole numbers, both positive and negative
- `int`'s are finite (why?), i.e. they have a limited range that is implementation dependent
- `int` is short for integer
- Examples of `int`'s are: 123, -23, 0, 2352
- An `int` without a sign (+ or -) is assumed to be positive
- 2,353 is not an `int`, 2353 is an `int`
- What operations can be performed on integers?

09/03/04

CS150 Introduction to Computer Science 1

8

## `double`

- The `double` data type is used to store real numbers, both positive and negative
- Real numbers can contain fractional parts
- `double`'s are finite
- Examples of `double`'s are: 1.0, -2.3, -.3, 12E5, -1E-2
- A `double` without a sign (+ or -) is assumed to be positive
- 2,353.99 is not a `double`, 2353.99 is a `double`

09/03/04

CS150 Introduction to Computer Science 1

9

## `char`

- The `char` data type is used to store single characters (letters, digits, special characters)
- `char` values are enclosed in single quotes
- Examples of `char`'s are: `'A'`, `'a'`, `'*'`, `'2'`, `'$'`

09/03/04

CS150 Introduction to Computer Science 1

10

## Examples

- Remember, the format for declaring variables is:
  - `data-type identifier;`
- You can declare variables of the different data types as follows
  - `int num1;`
  - `double num2;`
  - `char letter;`

09/03/04

CS150 Introduction to Computer Science 1

11

## Identifiers

- C++ does place limits on what names you can call your variables
- Rules
  - Identifiers must begin with a letter or an underscore
  - Identifiers must consist of letters, numbers and underscore, nothing else
  - Identifiers cannot be a reserved keyword

09/03/04

CS150 Introduction to Computer Science 1

12

## Reserved Keywords

- These are words that are reserved by C++ to implement various features
- Examples of keywords that we have seen so far are `int`, `double`, `const`, `return`
- A list of C++ keywords can be found on page 75 of your textbook

09/03/04

CS150 Introduction to Computer Science 1

13

## Identifiers

- Identifiers are case sensitive
  - `int num1`;
  - `int Num1`;
  - `num1` and `Num1` are different variables
- You should always try to use meaningful variable names
- If you have a variable that represents the width, then call it `width` not `w`

09/03/04

CS150 Introduction to Computer Science 1

14

## Identifiers

- Which of the following declarations are invalid and why?
  - `char Letter1`;
  - `char lletter`;
  - `double inches, kms`;
  - `double inches*num`;
  - `int joe's`;
  - `Int cent_per_inch`;
  - `double two-dimensional`;
  - `char hello`;
  - `int return`;
  - `size int`;

09/03/04

CS150 Introduction to Computer Science 1

15

## Variable Declarations

- All the variable declarations that we have seen are of the form
  - `data-type identifier`;
- This form declares a variable of the specific type, gives it the specific name (identifier) and allocates the space in memory to store the value of this variable
- However, no value has been assigned to this variable as yet

09/03/04

CS150 Introduction to Computer Science 1

16

## Variable Declarations

- When declaring multiple variables of the same type, it is preferable to place each variable declaration on a line along with a comment specifying the use of the variable

```
◦ double miles; // Distance in miles from user
◦ double kms; // Distance in kms
```

09/03/04

CS150 Introduction to Computer Science 1

17

## Constant Declarations

- Constants are declared by stating
  - `const`
  - Type of data (data type)
  - Name to identify the variable (identifier)
  - `=`
  - Value assigned to this constant that will never change
  - Semicolon (;)
  
- `const data-type identifier = value`;
- `const double KM_PER_MILE = 1.609`;

09/03/04

CS150 Introduction to Computer Science 1

18

## Constant Declarations

---

- Constant values *never* change
  - KM\_PER\_MILE will always be 1.609
- In C++ we typically place constant declarations before any other declarations in the program

09/03/04

CS150 Introduction to Computer Science 1

19

## Example

---

- Can you spot what is incorrect in the following program:

```
int main()
{
    const int pi = 3.14;
    double num;
    int i, j;

    num = e2;
    i = 4,000;
    ch = "b"; j = i;
    pi = 5;

    return 0;
}
```

09/03/04

CS150 Introduction to Computer Science 1

20

## Problem

---

- The problem specified at the end of class on Wednesday required us to write a program to calculate the area of a circle.
- What constant declarations does our program need?
- What variable declarations does our program need?

09/03/04

CS150 Introduction to Computer Science 1

21

## Summary

---

- In today's lecture we discovered
  - How Data that is used by a program can be declared and stored
  - The difference between constants and variables
  - What constitute valid identifier names
  - The three primitive data types; int, double, char
- We have covered p. 26 - 31 of your textbook

09/03/04

CS150 Introduction to Computer Science 1

22