

# Assignment 2

---

**Student Name: Xiangdong Yu**  
**Student UIN: 722001506**

Grader Name:  
Grader UIN:

---

**Reading Assignment:** C++ Primer, 5th edition

- Chapter 2 - Variables and Basic Types

**True or False:**

1. **0.5 pt** - C++ is a statically typed language; type checking is done at compile time. **(T)**
2. **0.5 pt** - Two string literals that appear adjacent to one another and that are separated only by spaces, tabs, or newlines are concatenated into a single literal. **(T)**
3. **0.5 pt** - All objects must be explicitly initialized. **(F)**
4. **0.5 pt** - Identifiers in C++ are not case-sensitive. **(F)**

**Short Questions:**

1. **1 pt** - Describe the difference between initialization and assignment.

Initialization: A newly created object is first given a value. There is no previous value to overwrite (destroy).

Assignment: An existing object is given a different value (copied from another object). The previous value of the existing object is overwritten (destroyed).

Assignment can be done as many times as desired whereas initialization can be done only once.

2. **1 pt** - What is the scope of a name?

A scope is a part of the program in which a name has a particular meaning. Most scopes in C++ are delimited by curly braces. The same name can refer to different entities in different scopes. Names are visible from the point where they are declared until the end of the scope in which the declaration appears.

In computer programming, the scope of a name binding – an association of a name to an entity, such as a variable – is the part of a computer program where the binding is valid: where the name can be used to refer to the entity.

3. **1 pt** - Determine the types and values of each of the following variables.

(a) `int * ip, i, &r = i;`

`ip` is a pointer to int, while `i` is an int; `r` is a reference to int `i`.

(b) `int i, *ip = 0;` (in a class as opposed to a function)

`i` is an int assigned value 0, while `ip` is a valid null pointer.

(c) `int* ip, ip2;`

`ip` is a pointer to int, and `ip2` is a solely int.

4. **1 pt** - Describe the difference between a pointer to a constant and a constant pointer.

A constant pointer is a pointer that cannot change the address its holding. In other words, we can say that once a constant pointer points to a variable then it cannot point to any other variable. A constant pointer is declared as follow: `<type of pointer> * const <name of pointer>`.

As evident from the name, a pointer through which one cannot change the value of variable it points is known as a pointer to constant. These types of pointers can change the address they point to but cannot change the value kept at those address. A pointer to constant is defined as: `const <type of pointer>* <name of pointer>`.

### **Programming Challenge:**

1. Construct a class named SpeedData using struct. This class should contain three data members: a string named scale, a double named speed, and an int named year. The scale should be mph or km/h.

Write a first application that creates an instance of SpeedData using stdin, and then returns the speed on stdout in the two aforementioned formats. Write a second application that takes two speeds from stdin and returns the smallest of the two.

2. Implement your application in C++.
3. Commit your code as a CMake project on GitHub in a directory labeled Assignment2.