FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

Classes and Objects

CSPROG2

Computer Programming 2 for CS

FEU INSTITUTE OF TECHNOLOGY
COLLEGE OF ENGINEERING • COLLEGE OF COMPUTER STUDIES

CLASSES AND OBJECTS

What are the types

What classes and objects are

How to define a new class and create objects of that class

Objectives

In this lesson we will learn:

- What classes and objects are
- How to define a new class and create objects of that class
- What member functions and member data are
- What constructors are and how to use them

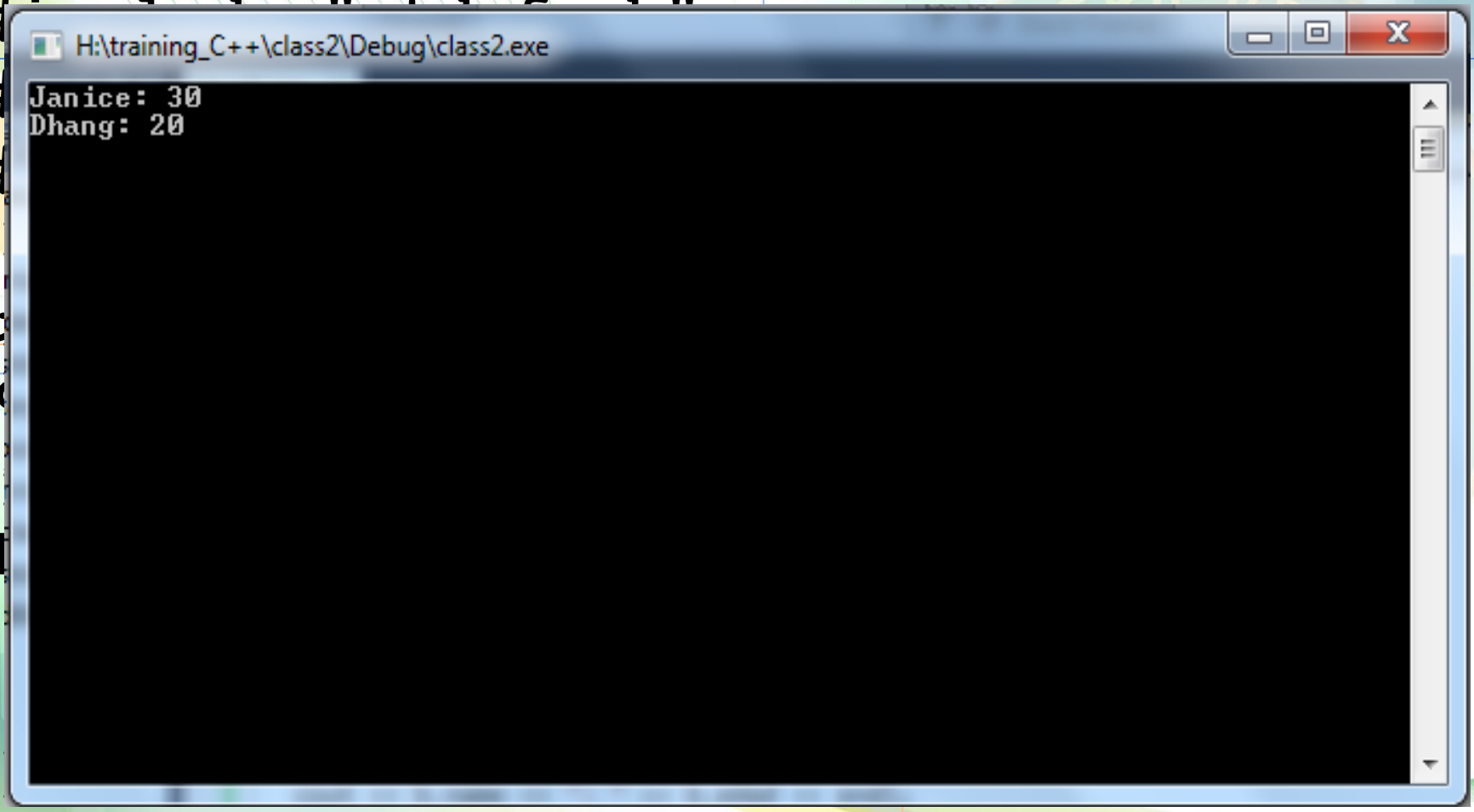
Basic declaration and member variables

```
struct person
{
  string name;
  int edad;
};
```

```
class person
{
public:
string name;
int edad;
};
```

Classes are declared with the **class** or **struct** keyword. Declaration of members are placed within this declaration.

```
//Declaration
#
#
in
{#
u
(
I
:
return 0;
}
```



*What is **type**?*

- Is an object with data and a set of abilities (such as int, long and float numbers)

The ***type*** of a variable tells you several things:

- Its size in memory
- The information it can hold
- The actions that can be performed on it

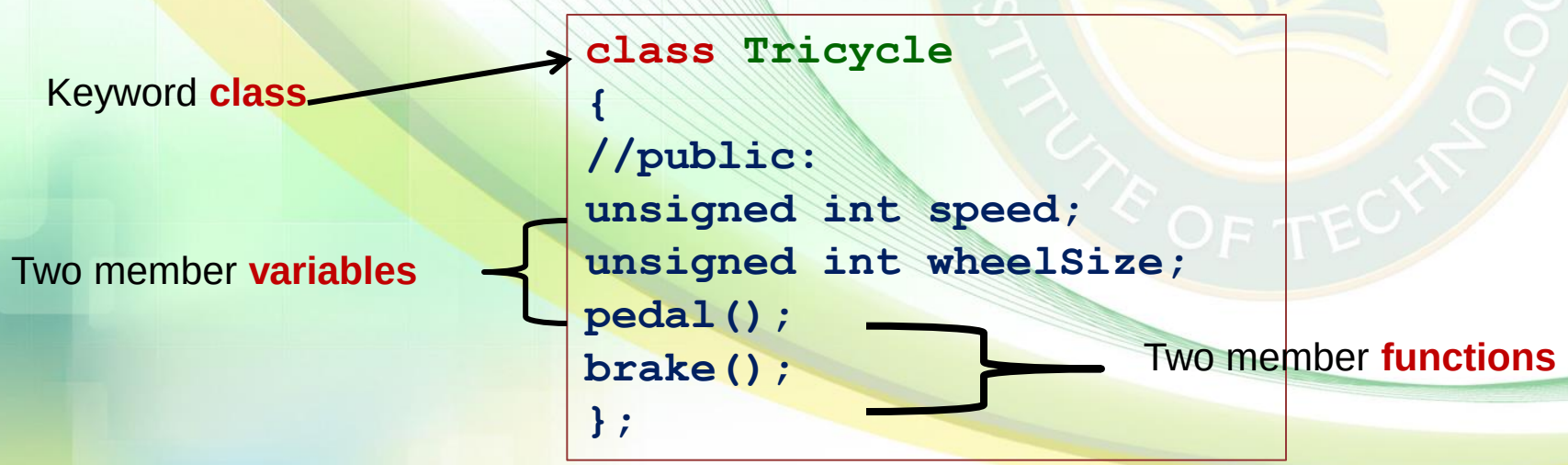
Classes and Members

A C++ class is a template used to create objects. Once you define a class, objects created from that class can be put to use like any other data type.

A class is a collection of related **variables** and **functions** bundled together. The variables can be of any type, including classes.

Variables make up the data in the class, and functions performs tasks using the data. Building these together is called ***encapsulation***.

To declare a **class**



Encapsulation of a class makes it possible for other programs to use the class without knowing how it works. Users of your class only need to know what it does, not how it does it.

Variables of the class are called *member variables*.

Member variables, also known as *data members* or *instance variables*, are part of your class, just like the wheel and brake are part of a tricycle.

The functions in the class use and modify the member variables. They are called the *member functions (or methods)* of the class. Member functions of the Tricycle class might include *pedal()* and *brake()*.

Defining an Object

An object is created from a class by specifying its class and a variable name.

Example:

```
Tricycle waley;
```

instantiate, meaning an object is just an individual instance of a class.

Accessing Class Members

Use the dot (.) operator to access the member functions and variable of the objects.

Example:

```
Tricycle waley;
waley.speed;
```

After the member function pedal() function has defined, the dot operator is used to call it:

Example:

```
waley.pedal();
```

Private Versus Public Access

The *public* keyword makes parts of the class variable and two public member functions. The public keyword makes these parts of the class available to the public—in other words, other classes and programs that use Tricycle objects.

All member variables and functions are private by default. *Private* members can be accessed everywhere else.

Example:

```
class Tricycle
{
    unsigned int speed;
    unsigned int wheelSize;
    pedal();
    brake();
}
```

When the public keyword appears in a class definition, all member variables and functions after the keyword are public.

```
class Tricycle
{
    int model = 10;
public:
    unsigned int speed;
    unsigned int wheelSize;
    pedal();
    brake();
}
```

The preceding code declares everything in the Tricycle class public except for the model member variable.

There is also a private keyword to make all subsequent member variables and functions private.

Example:

```
class Tricycle
{
public:
    int getSpeed();
    void setSpeed(int speed);
    void pedal();
    void brake();
private:
    int speed;
};
```

Implementing Member Functions

A member function definition begins with the name of the class followed by the scope resolution operator (**::**) and the name of the function.

Example:

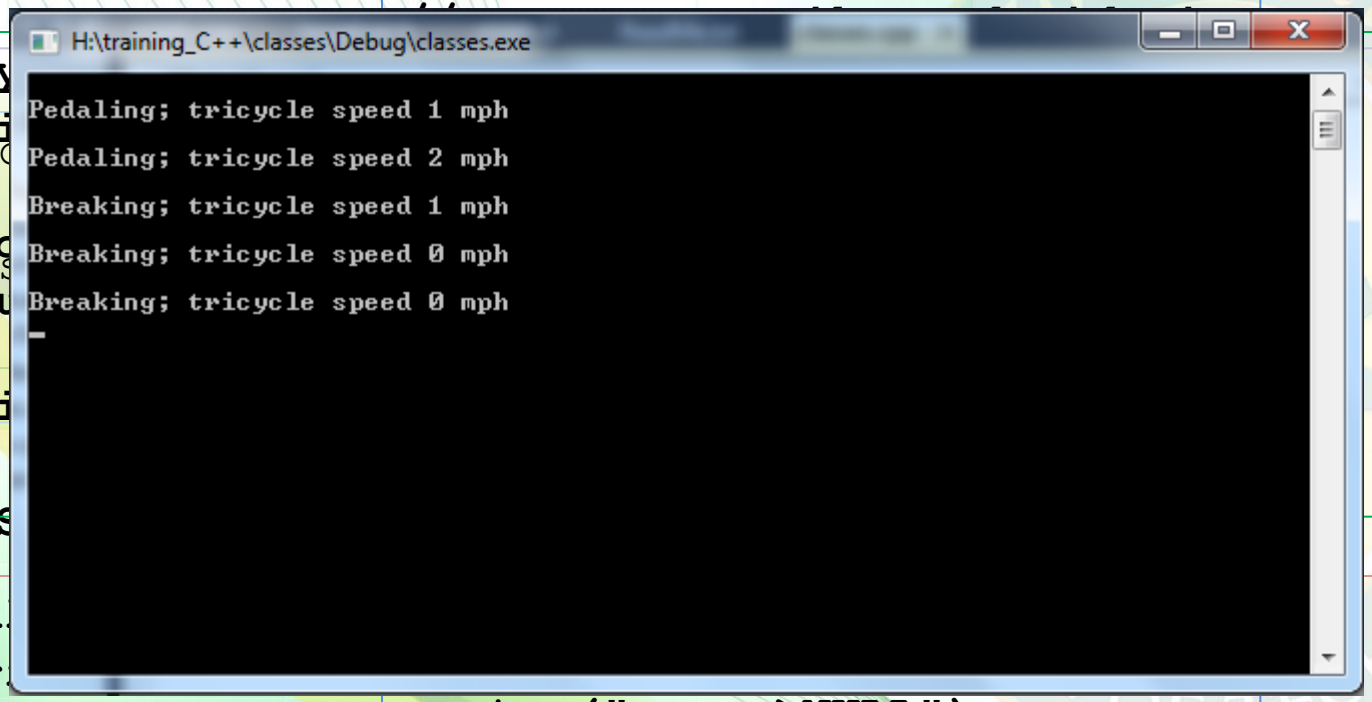
```
std::cout <<"\nPedaling; tricycle speed "<<speed<<"mph\n";
```

Scope Resolution Operator

```
// classes.cpp : Defines the entry point for the console
// application.
// Date created: April 26, 2011

#include <iostream>

// apply
void Tri
int Tric
{
{
setSpeed
return s
std::cou                                                    " mph\n";
}
}
void Tri
{
if (newS
{
// peda
speed =
void Tri
}
{
}
setSpeed(speed + 1);
std::cout <<"\nPedaling; tricycle speed " << speed << " mph\n";
}
```




```
system("pause>NULL");
return 0;
}
```

DO and DON'T

- DO use the keyword class to declare a class.
- DO use the dot operator (.) to access class members and functions.
- DON'T confuse a declaration with a definition. A declaration says what a class is. A definition sets aside memory for an object.
- DON'T confuse a class with an object.
- DON'T assign values to a class. Assign values to the data members of an object.

Constructors

When an object of file is created, C++ calls the constructors for that class. If no constructor is defined, C++ invokes a default constructor, which allocates a memory for the object, but does not initialized it.

Why define a constructor?

Uninitialized member fields have *garbage* in them. This creates the possibility of a serious bug (eg, an uninitialized pointer, illegal values, inconsistent values, ...).

Making Member Data Private

- To access private data in a class, you must create public functions known as **`accessor methods`**. *(*Use these methods to set and get the private member variables).
- These **`accessor methods`** are the member functions that other parts of your program call to get and set your private member variables. A public accessor method is a class member function used either to read (get) the value of a private class member variable or to set its value.

Declaring a constructor

- A constructor is similar to a function, but with the following differences. No return type.
- No return statement.

The Syntax of Derivation

When you create a class that inherits from another class in C++, in the class declaration you put a colon after the class name and specify the access level of the class (public, protected, or private) and the class from which it derives.

Access control is covered later. For now, you use public, as in this example:

```
class Dog : public Mammal
```