Lecture 4: Functions

Curtin FIRST Robotics Club (FRC) Pre-season Training

Scott Day

265815F@curtin.edu.au

November 11, 2016

Curtin University

Functions

In C++ we can subdivide the functional features of a program into blocks of code known as functions. In effect these are subprograms that can be used to avoid the repetition of similar code and allow complicated tasks to be broken down into parts, making the program modular.

Until now you have encountered programs where all the code (statements) has been written inside a single function called main(). Every executable C++ program has at least this function. In the next sections we will learn how to write additional functions.

Table of contents

Function Justification

t

t

t

Function Definition

In C++, a function is a group of statements that is given a name, and which can be called from some point of the program. The most common syntax to define a function is:

```
returnValueType functionName(type parameter1, type parameter2, ...)
{
    statements;
}
```

Where:

return-value-type Is the type of the value returned by the function.

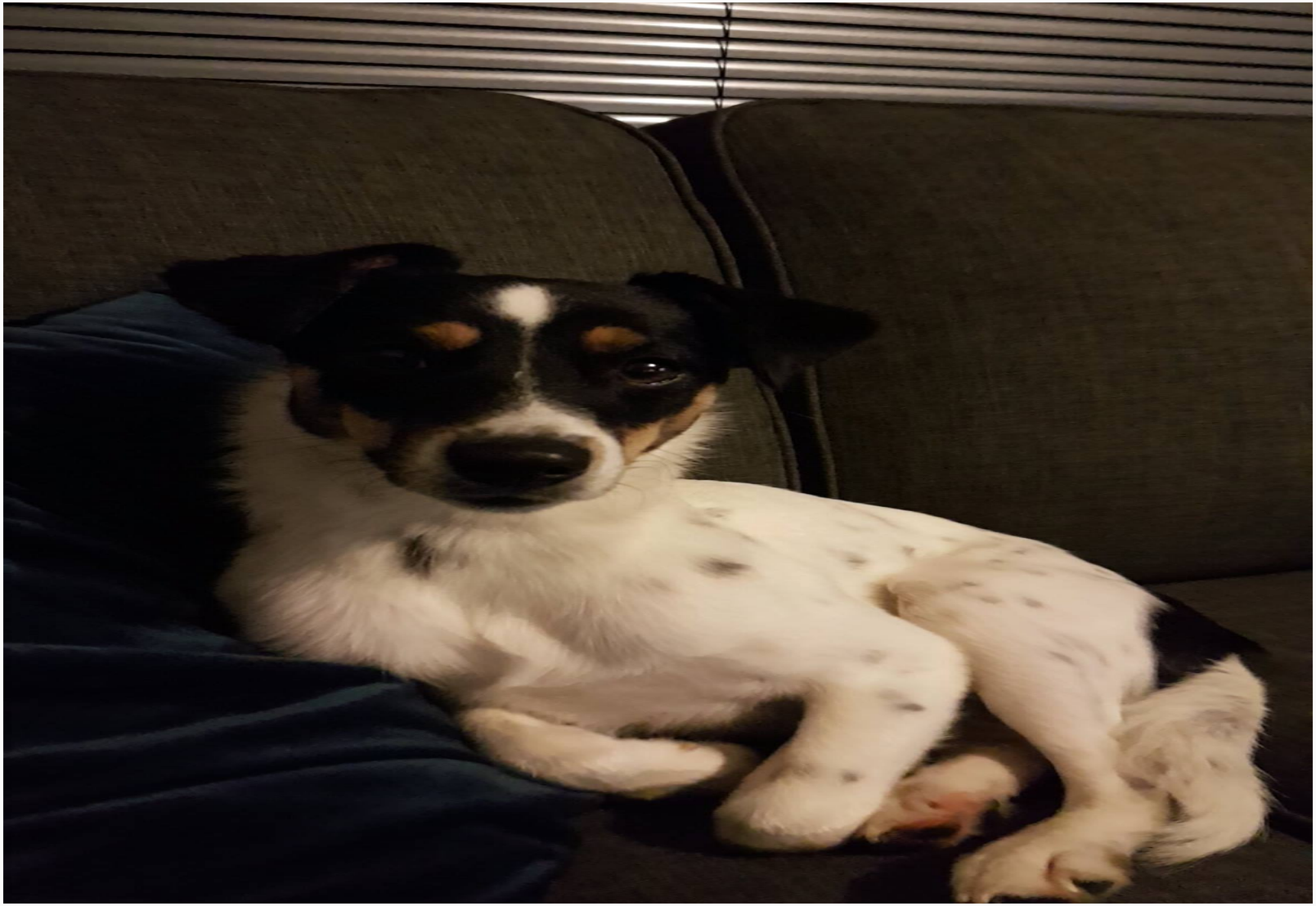
function-name Is the identifier by which the function can be called.

parameters Each parameter consists of a type followed by an identifier, with each parameter being separated from the next by a comma.

statements Is the function's body. It is a block of statements surrounded by braces { } that specify what the function actually does.

t

Cute Dog



Example of Function Definition, Declaration and Call

t

t

t

Function Header and Body

t

t

t

Function Declaration

t

t

t

Function Call and Execution

t

t

t

Function Arguments

t

t

t

Passing by Value or Reference

t

t

t

Recursion - just to f*** with you

t

t

t

References I