Details: We look at functions in several languages, with a particular focus on functions as “first-class objects.”

What is a “First-Class Object”? 

According to Scott, a first-class value in a programming language is a value that can be passed as a parameter, returned from a subroutine, and assigned to a variable (chapter 3, p. 154). Obviously, things like int, double, char, etc., are first-class objects in C and Java, arrays are first-class objects, user-defined objects are first-class objects, . . . . What about functions? Can a C or Java function take a function as a parameter? Can it return one? Can we assign a function to a variable? NOTE: we are talking about the function definitions themselves, not the results of calling the functions. Obviously we can do things like “x = f(y);”. But can we also do “z = f;”? 

Functions as Variables in C 

In C, the following declaration declares variable f to be a function with one int argument that returns an int value:

```
int (*f)(int);
```

Similar notations are used for other kinds of functions, e.g.,

```
double (*g)(double);
```

is a function with one double argument that returns a double;

```
char (*h)(char[],int);
```

is a function with two arguments—a char array and an int—that returns a char. Sample program fvar.c illustrates these examples and shows how we can assign values to f, g, and h.

We can pass a function to another function through the parameter list using the same notation as above.

```
double avg(int (*f)(int), int a, int b)
```
The first parameter in function `avg`, above, is a function from `int` to `int`. Sample `fpar.c` shows an example of using function `avg` with several different functions as arguments.

1. **[Text file answer required.]** Think of a concrete application for function parameters—in what real-life programming task might we find it helpful to pass a function into another function as a parameter?

2. **[Try it yourself.]** Imitating the style of program `fpar.c`, write a C program that has a function variable named “f” that takes a `double` and returns an `int`. Create three functions (I’ll call them `f1`, `f2`, and `f3`), different from `sqr`, `cube`, and `weird`, that “do something” with a `double` and return an `int`. (Examples: round the `double` value to an `int` or find the power of 2 that is nearest to the `double` value.) Write one more function named `calc` that takes a function parameter of the type just described and calculates something with it (you can include other parameters in addition to the function parameter as I did in my `avg` function in program `fpar.c`). Test your `calc` function with each of the three functions `f1`, `f2`, and `f3`.

Be sure to document each of your functions as well as providing appropriate summary information in header comments.

### Java Lambda Expressions

In Java version 8 a new structure was introduced: lambda expressions. A lambda expression describes a function; for example,

```java
(Double x)->3+2*x-4*x*x
```

defines a function with one `Double` argument, `x`, that returns the value shown on the right.

A new class in Java named “Function” can “apply” such a function to an argument. Program `FunParam.java` shows an example of this.

3. **[Text file answer required.]** Note that, unlike the earlier C program example, the functions passed as parameters are not already-existing functions with names (like “`sqr`” or “`cube`”) but “nameless” functions. These could even be constructed at run time using variable values in place of the constants.

Think of a concrete application for lambda expressions—in what real-life programming task might we find it helpful to create such “anonymous” functions at execution time?

4. **[Try it Yourself.]** Write a Java program modeled after `FunParam.java`. Use the `Function<...,...>` class to pass in a lambda expression as a parameter to another function and do “something interesting” by applying this function. Try a `Function<Integer,Integer>` or `Function<String,String>` function.

Save your programs, comment them, and upload the files to your repository. Also upload your answers to the questions.