Local Variables and Function Arguments Solutions

Local variables

- Explain what is meant by "local variable" in C++

  - A local variable exists within a lexical scope which is bounded by braces {}. The memory used to store it is automatically allocated on the program's stack when needed and released at the end of the scope. The variable will be initialized with an initial value, if one is given. Otherwise, it will be default initialized

- In the following code, what is the initial value of d?

  ```
  double d;
  ```

  - As no initial value is given, d will be default initialized. It is a built-in type, so its initial value is undefined

Argument Passing

- Explain what is meant by the following terms:

- Pass by value

  - The value passed by the caller is copied into the function's argument. The argument in the function body will be an independent object which only exists during the function call. Any changes made to this object will not affect the caller

- Pass by address

  - The caller passes the address of an object. The argument in the function body will be a pointer which has the same address. Any changes made through the pointer will also be made to the caller's object

Argument Passing

- Pass by reference
  - Similar to pass by address, but the addressing and dereferencing is done implicitly. The caller appears to pass the object by value; the argument in the function body will be a reference to this object. Any changes made to the reference will also be made to the caller's object

Function Arguments

- Write programs which demonstrate
  - Pass by value
  - Pass by address
  - Pass by reference
- Add code to your programs to show the addresses of the variables involved
- Explain your observations
  - Pass by value - the variables have different addresses, because they are independent objects
  - Pass by address and pass by reference - the caller's variable and the argument in the function body represent the same object