Name: CAELAN KIMBALL **ID: 020022**
Date: 10/21/2019 **Class: Computer Programming 2**
Score: 20 / 25 (80%)

CP2 Chapter 6 Pre-Test

TRUE/FALSE

✗ T 1. True/False: A local variable and a global variable may not have the same name within the same program.

Points: 0 / 1

✓ T 2. True/False: It is possible for a function to have some parameters with default arguments and some without.

Points: 1 / 1

✓ F 3. True/False: A function's return data type must be the same as the function's parameter(s).

Points: 1 / 1

✓ F 4. True/False: Local variables are initialized to zero by default.

Points: 1 / 1

✓ T 5. True/False: It is not considered good programming practice to declare all of your variables globally.

Points: 1 / 1

✓ T 6. True/False: You may use the `exit()` function to terminate a program, regardless of which control mechanism is executing.

Points: 1 / 1

MULTIPLE CHOICE

✓ D 7. This is a collection of statements that performs a specific task.

a. infinite loop
b. variable
c. constant
d. function
e. None of these

Points: 1 / 1

✓ D 8. A function is executed when it is:

a. defined
b. prototyped
c. declared
d. called
e. None of these

Points: 1 / 1

✗ D 9. The value in this type of local variable persists between function calls.

a. global
b. internal
c. static
d. dynamic
e. None of these

Points: 0 / 1

✓ B 10. These types of arguments are passed to parameters automatically if no argument is provided in the function call.

a. Local
b. Default
c. Global
d. Relational
e. None of these

Points: 1 / 1

✓ A 11. When used as parameters, these types of variables allow a function to access the parameter's original argument.

a. reference
b. floating-point
c. counter
d. undeclared
e. None of these

Points: 1 / 1

✗ A 12. This statement causes a function to end.

a. `end`
b. `terminate`
c. `return`
d. `release`
e. None of these

Points: 0 / 1

✓ B 13. ________ functions may have the same name, as long as their parameter lists are different.

a. Only two
b. Two or more
c. Zero
d. Un-prototyped
e. None of these

Points: 1 / 1

✓ D 14. This is a statement that causes a function to execute.

a. `for` loop
b. `do-while` loop
c. function prototype
d. function call
e. None of these

Points: 1 / 1

✓ B 15. A function ________ eliminates the need to place a function definition before all calls to the function.

a. header
b. prototype
c. argument
d. parameter
e. None of these

Points: 1 / 1

✓ B 16. If a function does not have a prototype, default arguments may be specified in the function ________.

a. call
b. header
c. execution
d. return type
e. None of these

Points: 1 / 1

✓ B 17. This is a dummy function that is called instead of the actual function it represents.

a. `main` function
b. stub
c. driver
d. overloaded function

Points: 1 / 1

✓ B 18. What is the output of the following program?

```
#include <iostream>
using namespace std;

void showDub(int);

int main()
{
    int x = 2;

    showDub(x);
    cout << x << endl;
    return 0;
}

void showDub(int num)
{
    cout << (num * 2) << endl;
}
```

a. 2
2

b. 4
2

c. 2
4

d. 4
4

Points: 1 / 1

✗ A 19. What is the output of the following program?

```
#include <iostream>
using namespace std;

void doSomething(int&);

int main()
{
    int x = 2;

    cout << x << endl;
    doSomething(x);
    cout << x << endl;
    return 0;
}

void doSomething(int& num)
{
    num = 0;
    cout << num << endl;
}
```

a. 2
0
2

b. 2
2
2

c. 0
0
0

d. 2
0
0

Points: 0 / 1

✗ A 20. Which line in the following program contains the header for the `showDub` function?

```
1   #include <iostream>
2   using namespace std;
3
4   void showDub(int);
5
6   int main()
7   {
8       int x = 2;
9
10      showDub(x);
11      cout << x << endl;
12      return 0;
13  }
14
15  void showDub(int num)
16  {
17      cout << (num * 2) << endl;
18  }
```

a. 4
b. 6
c. 10
d. 15

Points: 0 / 1

✓ B 21. Look at the following function prototype.

```
int myFunction(double);
```

What is the data type of the function's parameter variable?

a. `int`
b. `double`
c. `void`
d. Can't tell from the prototype

Points: 1 / 1

✓ A 22. Look at the following function prototype.

```
int myFunction(double);
```

What is the data type of the function's return value?

a. `int`
b. `double`
c. `void`
d. Can't tell from the prototype

Points: 1 / 1

✓ C 23. Look at the following function prototype.

```
int myFunction(double, double, double);
```

How many parameter variables does this function have?

a. 1

b. 2

c. 3

d. Can't tell from the prototype

Points: 1 / 1

✓ C 24. What is the output of the following program?

```
#include <iostream>
using namespace std;

int getValue(int);

int main()
{
    int x = 2;

    cout << getValue(x) << endl;
    return 0;
}

int getValue(int num)
{
    return num + 5;
}
```

a. 5

b. 2

c. 7

d. "getValue(x)"

Points: 1 / 1

✓ B 25. Here is the header for a function named `computeValue`:

```
void computeValue(int value)
```

Which of the following is a valid call to the function?

a. `computeValue(10)`

b. `computeValue(10);`

c. `void computeValue(10);`

d. `void computeValue(int x);`

Points: 1 / 1