

# LOUIS MENSAH BONSU

CLASS

Name: .....

Index No: .....

## CIRCLE THE CORRECT ANSWERS (ANSWER ALL)

1. Select from the list below the choice that is NOT a relational operator.

☒ A. ||  
☒ B. ==  
☐ C. >  
☐ D. !=

2. Which of the symbols below is not a relational operator?

☒ A. =  
☐ B. >  
☐ C. !=  
☐ D. ==

3. Which of the symbols below is not a logical operator?

☐ A. &&  
☒ B. <  
☐ C. !  
☐ D. ||

4. What is wrong with this code?

```
if(x = 2)
{
    cout<<"The number is 2";
}
```

- A. There should not be { }'s  
 B. A semicolon is missing in the if statement `if(x=2)`  
☒ C. The if statement should be `if ( x == 2 )`  
 D. Nothing

5. What is the purpose of a relational operator in C++?

- A. It is used with pointers to access addresses.  
☒ B. It is needed to do comparisons between variables.  
 C. The cout statement must have relational operator(s).  
 D. The program uses it to assign values to variables.

6. Which if statement would be true if you needed to see if x (an int) is between 0 and 50 inclusive?

A. `if( x >= 0 && x < 50 )`  
☒ B. `if( x >= 0 && x <= 50 )`  
 C. `if( x >= 0 || x <= 50 )`  
 D. `if( x > 0 && x < 50 )`

7. Which if statement would be true if you needed to see if x (an int) is the same as 34?

☒ A. `if( x == 34 )`  
 B. `if( x <= 35 )`  
 C. `if( x = 34 )`  
 D. `if( x != 33 && x != 35 )`

8. In this while loop statement, `while(counter < 10)`

the variable counter is an int. Which statement below is an equivalent way to write this while statement?

- A. `while(10 > counter)`  
 B. `while( counter <= 9)`  
 C. `while(9 > counter)`  
☒ D. A and B are correct

9. How many times will this for loop execute? (Note: i is an int.)

```
for(i = 200; i > 0; --i)
```

- A. 0, the loop will not execute  
☒ B. 199  
☒ C. 200  
 D. The loop is not written correctly. It would not compile.

10. How many times will this for loop execute? (Note: i is an int.)

```
for(i = 0; i < 5; --i)
```

- A. 0, the loop will not execute  
 B. 9  
 C. 10  
☒ D. It is an infinite loop.

11. How many times will this for loop execute? (Note: i is an int.)

```
for(i = 0; i < 20; i = i+2)
```

- ☒ A. 5  
☒ B. 10  
 C. The loop is not written correctly. It would not compile.  
 D. It is an infinite loop.

12. What keywords are used to construct a switch statement?

- A. case, jump, break, default  
☒ B. switch, case, break, default  
 C. break, default, case, goto  
 D. switch, case, break, else

13. What keywords are used to construct if statements?

- A. else  
☒ B. if, else  
 C. if, else, case  
 D. if, else, endif

14. What is the output of this program?

```
int x = 10, y = 12;
if(x > y)
```

```
{
    cout<<" x is greater than y.";
}
```

- A. x is greater than y  
 B. false  
☒ C. Crash. The statement is written incorrectly.  
☒ D. Nothing. The statement inside the braces will not execute.

15. This if statement should assign the heavier weight to heaviest and the lighter weight to lightest. What is wrong with this code?



1 if (weight1 > weight2)  
 2 heaviest = weight1;  
 3 lightest = weight2;

- A. Nothing. It works fine.  
 B. Heaviest is weight1 regardless of the if statement.  
 C. The statement is written incorrectly - crash.  
 D. Line 3 gets executed no matter what. {} are needed.

16. What is wrong with this while loop?

```
int count = 0;
while(count < 50)
{
    cout<<endl<<count;
}
```

- A. Nothing.  
 B. Infinite loop.  
 C. It never gets executed.  
 D. it cout's "endl" 50 times.

17. Where is a global variable defined?

- A. Outside a function  
 B. Inside a function  
 C. Anywhere  
 D. In a function header line

18. Given this code:

```
int z;
int main( )
{
    int y = 3, x = 2;
    z = x + y;
    return 0;
}
```

What is the scope of variable z?

- A. local  
 B. static  
 C. global  
 D. none of the above

19. The list of input datatypes and variable names found in the function header line is referred to as the function's

- A. format.  
 B. arguments.  
 C. return types.  
 D. objects.

20. Which function name below is invalid?

- A. setName  
 B. CalculateAverage  
 C. Names  
 D. both B and C

21. If a variable is declared as static inside a function, what

- parts of the program can see the variable?  
 A. Any function called after the static variable was declared.

- B. Only the function in which it was declared.  
 C. The entire program.  
 D. Other functions declared as static.

22. At what point is a static variable released from memory?

- A. When the program terminates.  
 B. After the return statement is executed.  
 C. When the function terminates.  
 D. When the value changes.

23. What is wrong with this call statement?

```
PrintName(string name);
```

- A. It's missing "void".  
 B. The datatype should be removed.  
 C. The semicolon should be removed.  
 D. Nothing is wrong with it.

24. Which function name is invalid?

- A. Function-to-Calc-Price  
 B. Function to CalcPrice  
 C. Function\_To\_Calc??  
 D. They are all invalid

25. Which function name is valid?

- A. PrintPrice4Me  
 B. Print??  
 C. 4PrintingPrice  
 D. They are all valid

26. What is true about a function call statement?

- A. Every function must be called at least once.  
 B. A function may be called whenever it is needed.  
 C. A function may be called only one time in a program.  
 D. A function may not be called from separate functions.

What is wrong with this code?

```
int Money4Me( int dollars )
{
    int pennies, coins;
    coins = dollars * 100;
}
```

- A. The function name is invalid.  
 B. There is no return type.  
 C. You can't have math statements inside a function.  
 D. Nothing is wrong with it.

27. What is wrong with this code?

```
void PrinttoScreen(int dollars, int cents)
{
    cout << "\n The dollars are << dollars;
    cout << "\n The cents are << cents;
    return OK
}
```

- A. The return type is not void.  
 B. You can't return an OK.  
 C. Both A & B  
 D. Nothing is wrong with it.



42/08 HA2/5308 21001

28. What is wrong with this code?

```
int ShowMeTheMoney()  
{  
    int cents, dollars;  
    cout << "\n Enter dollars and cents";  
    cin >> dollars, cents;  
    return (dollars, cents);  
}
```

- A. You can't have a comma in the cin statement.
- B. You can't have a comma in the return statement.
- ☒ C. Both A & B.
- D. There is nothing wrong with it.

29. What is wrong with this code?

```
float CalcCost( int unit )  
{  
    float cost;  
    cost = 5.59 * unit;  
    return cost;  
}
```

- A. The function name is invalid.
- B. The return type should be int.
- ☒ C. You can't have math statements inside a function.
- D. Nothing is wrong with it.

\* 30. What is the purpose of the function prototype?

- A. Declares variables that need to be used.
- B. Provides the compiler with function information.
- C. Allows the function to be called.
- D. You don't always need a prototype.

31. What is the purpose of the function call?

- ☒ A. Control is passed to the called function.
- ☒ B. The program is passed all of the called function's variables.
- C. main() gets to use the function's variables.
- D. Both A & C.

\* 32. Why doesn't the main function need a prototype statement?

- ☒ A. It does have one, we just don't see it.
- B. It gets called by value
- C. Because it has a return type.
- ☒ D. The operating system knows to look for it.

33. After a function executes, to where does control of the program return?

- A. To main()
- B. To the called function.
- C. To the last line where it left off.
- ☒ D. To the call statement.

34. Another word for "local" variables is \_\_\_\_\_ variables.

- ☒ A. static
- ☒ B. automatic
- C. initialized
- D. function

\* 35. Why is it not necessary to write a function prototype and function header line/body when using a function from a standard library?

- A. You have to use the prototype and function header line.
- B. The prototype isn't defined until the code is compiled/
- C. The library substitutes compiled code for the linker.
- ☒ D. The library defines the function, so the program can just use it.

36. What is a variable's scope?

- A. The length of time the variable is in existence.
- B. Determines who can see and access the variable.
- ☒ C. Both A & B
- D. None of these.

37. Why is it not necessary to pass a global variable to a function?

- A. You have to pass global variables to functions.
- B. main() passes them for you.
- ☒ C. All functions can see and access global variables.
- D. They remain local to main().

38. Why is it not necessary to return a variable from a function

if the variable is declared as global?

- A. You have to return all variables regardless of scope.
- ☒ B. All functions can see and access global variables.
- C. main() gets copies of global variables.
- D. Each function gets its own copy of global variables.

39. In the function header line and body:

```
void CalcArea(double rad) {  
    double rad, area;  
    area = PI * pow(rad, 2);  
    cout >> "The area is==>" << area << endl;  
}
```

Assume the variable PI is declared globally.

Given what is shown here, what would be the compiler error?

- A. There is no error.
- B. rad is declared twice.
- ☒ C. The cout statement is incorrect.
- D. The pow function is misspelled.

40. What is wrong with this code?

```
int ConvertFtToInches( int ft )  
{  
    int inches;  
    inches = ft * 12;  
}
```

- A. The function name is invalid.
- B. You can't have the variable name in the input argument list.
- ☒ C. There is no returned value.
- D. It will return the wrong value.



# LOUIS MENSAH BOANSU

## SECTION B:

Answer Question 1 & 2 on the Question Paper (Space Provided)

1. One large chemical company pays its salespeople on a commission basis. The salespeople each receive \$200 per week plus 9 percent of their gross sales for that week. For example, a salesperson who sells \$5000 worth of chemicals in a week receives \$200 plus 9 percent of \$5000, or a total of \$650. Develop a C++ program inputs each salesperson's gross sales for last week and calculates and displays that salesperson's earnings. Process one salesperson's figures at a time.

ANS:

```
#include <iostream>
using namespace;

int main ( )
{
    double Sales, P = 0.09, TotalBarn;
    cout << "Calculating Sales earnings";
    cout << "Please enter total sales";
    cin >> Sales;
    TotalBarn = Sales * P;
    TotalBarn = Sales * P + 200;
    cout << "The total gross sales is ";
    cout << TotalBarn;
}
```

2. Develop a simple function to calculate the area of Circle. The program should ask the user to input the radius.

ANS:

```
#include <iostream>
using namespace;

void circle ( )
{
    double radius, Area, C = 3.142;
    cout << "Area of a circle";
    cout << "Please enter the radius ";
    cin >> radius;
    Area = radius * radius * C;
    cout << "The area of the circle is ";
    cout << Area;
}

int main ( )
{
    cout
    area circle ( );
    return 0;
}
```