26

What is the output of the following code snippet?

```
vector<int> num(4);  // This sets size - not capacity!  The four elements are initialized to 0
num.push_back(4);  // pushing a new element onto the vector increases the size by 1 and makes it 5
cout << num.size();  // the correct answer is in fact 5
```

1

3

4

5

25

Which of the following statements is true about data members of a class definition?

Data members are set in the constructor and cannot be changed later.

Every object of the defined class has its own set of data members, with possibly different values.

All data members are shared by all objects of the defined class.

The value stored in a data member can be changed by any function in the program.

24

In the given code snippet, what type of member function is view() ?

```
class CashRegister
{
public:
   void view() const;

private:
   int item_count;
   double total_price;
};
void CashRegister::view() const
{
   cout << item_count << endl;
   cout << total_price << endl;
}
```

Accessor member function

Mutator member function

Private member function

Constructor

23

Which of the following statements is correct about an accessor member function?

It modifies one or more data members of an object.

It returns the value of a data member of an object but does not modify any data member values.

It is only called by other member functions of an object.

It cannot be declared with the keyword const.

22

Which of the following statements about the public interface for the Car class is incorrect?

```
class Car
{
  public:
    void start();

  private:
    double speed;
    void stop();
};
```

This interface contains the start() function.

This interface does not contain the stop() function.

The code snippet includes the public interface for the class but lacks the definition of the member functions.

None are incorrect. All of the listed items are correct.

21

Based on this instruction:

```
char ch = 100;
```

which of the following is a legally correct way of obtaining the memory location of ch and printing it (the memory location) to standard output?

```
cout << ch << endl;
cout << *ch << endl;
cout << &ch << endl;
cout << *(&ch) << endl;
```

20

Why should a parameter be pass by reference?

To limit access to caller's memory from the called function.

To effectively return multiple values to the caller function.

To pass an argument by value.

To allocate new memory for the passed argument.

19

Which of the following statements is true about pointers?

Pointers contain values as well as addresses of variables.

Pointers contain address locations of variables.

Pointers do not have any address location of their own.

Pointers occupy permanent storage locations inside the hard disk.

18

A function prototype ...

is used as an initial example of a correct function signature.

specifies the function's name and type signature, but omits the function body.

creates multiple functions with different argument lists.

overloads an existing function to accept other argument types.

17

Which one of these statements is false:

Arrays are composite data types.

A function cannot change the dimensions of a two-dimensional array that is passed as a parameter.

Elements of different columns in a two-dimensional array can have different types.

Arrays can contain strings as elements.

16

Which of the following would correctly read an entire line from an input file stream named fin into a string variable named line?

fin.getline(line,80);

fin.getline(line);

getline(fin, line);

fin.getline(line,'\n\');

15

1 Point

Which is the proper way to determine how many characters are in the string variable named str?

str.getLength()

str.length()

length(str)

getLength(str)

14

1 Point

A call to a C++ function is:

the name of the function followed by empty parentheses

the name of the function followed by any number of arguments, regardless of the number of parameters in the definition

the name of the function followed by exactly the number and type of arguments as there are parameters in the definition

the name of the function only

13

In a switch statement, when a break statement is encountered, an immediate transfer of control is made to:

the default case of the switch statement

a goto statement

the else clause

the statement after the end of the switch statement

12

The condition that works for all cases in:
" A possibly empty string myString, which must start with an 'F' or a 'G' "
is:

A myString.empty() || ((myString[0] == 'F') || (myString[0] == 'G'))

B !myString.empty() || ((myString[0] == 'F') || (myString[0] == 'G'))

C myString.empty() && ((myString[0] == 'F') || (myString[0] == 'G'))

D !myString.empty() && ((myString[0] == 'F') || (myString[0] == 'G'))

11

Which, if any, of the following compound conditions can never be short-circuited by C++?

A

```
(my_char == 'R') || (their_car == 'r')
```

B

```
!my_string.empty() && (s[0] != 'c')
```

C

```
(my_char == 'a') != (their_char == 'b' )
```

D

```
(my_int < 5) || (their_int >= 6)
```

10

1 Point

Which of the following is the correct syntax for an if-else statement?

A

```
if (x < 10); {
  size = "Small";
}
else if (x < 20) {
  size = "Medium";
}
```

B

```
if (x < 10) {
  size = "Small";
}
else if (x < 20) {
  size = "Medium";
}
```

C

```
if (x < 10) {
  size = "Small";
}
else (x < 20) {
  size = "Medium";
}
```

D

```
if {
  size = "Small";
}
else if (x < 20) {
  size = "Medium";
}
```

A

B

C

D

9

The following function should sum all entered int values that are greater than 5, but it is incorrect. Find an error.

```
int main()
{
  int x, sum = 0;
  while (x < 10)
  {
    cin >> x;
    if (x > 5);
      sum = sum + x;
  }
  cout << "The sum of values > 5 is: " << sum << endl;
}
```

The while header needs a semicolon at the end of its line.

The semicolon at the end of the if statement is an error that the compiler should catch.

The semicolon at the end of the if statement causes all entered values to be summed.

The statement: sum = sum + x; should instead say: sum == sum + x;

8

What is the output of this program?

```
#include <iostream>
using namespace std;
int main()
{
  int x = 10;
  while (x > 0)
  {
    cout << x << " ";
    x = x + 3;
  }
}
```

10 13 16 19 22 25 . . .

0 3 6 9 12 15 18 21 24

The compiler will not compile because this has an infinite loop.

This will compile and will run without any problems.

7 was a survey

6

1 Point

Which one of the following is an assignment statement?

int a = 20;

assign 20 to a;

a = 20;

assign a = 20;

5

What is wrong with the following code snippet?

```
int price;
price = 9.42;
```

The price variable is never initialized

The data type for the price variable is not specified

There is a type mismatch between price and 9.42

The price variable is never assigned a value

4

A literal is all of these except:

a specific value.

type of variable.

of floating point type, if it contains a "."

a construct, whose type is determined by its syntax.

3

The type of a variable is ...

where in memory the variable is stored.

the value that is currently stored in the variable.

the set of all values for this variable

from where in the program the variable is visible.

2

What is the output of this program, if the user input is 2?

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
   string greeting = "Hello, ";
   int number = 0;
   int counter = 0;
   cout << "Please enter an integer: ";
   cin >> number;
   while(counter < number)
   {
      counter = counter + 1;
      cout << greeting;
   }
   cout << "Hello!";
}
```

greetings Hello!

greetings Hello, Hello!

Hello, Hello, Hello!

Hello, Hello!

1 was a survey