

FINALTERM EXAMINATION
Fall 2008
CS201- Introduction to Programming
Solved by vuZs Solution Team
Mateen, Farhat Qudsia
Zubair Hussain,Shoaib
www.vuzs.net/

Question No: 1 (Marks: 1) - Please choose one

There are mainly ----- types of software

- ▶ **Two**
- ▶ Three
- ▶ Four
- ▶ Five

www.vuzs.net/

Software is categorized into two main categories

- System Software
- Application Software P# 9

Question No: 2 (Marks: 1) - Please choose one

seekg() and write() are functionally _____.

- ▶ **Different**
- ▶ Identical
- ▶ Two names of same function
- ▶ None of the above

Question No: 3 (Marks: 1) - Please choose one

When a pointer is incremented, it actually jumps the number of memory addresses

- ▶ **According to data type**
- ▶ 1 byte exactly
- ▶ 1 bit exactly
- ▶ A pointer variable can not be incremented

“When a pointer is incremented, it actually jumps the number of memory spaces according to the data type that it points to”

Question No: 4 (Marks: 1) - Please choose one

www.vuzs.net/

setw is a parameterized manipulator.

- ▶ **True**

▶ False

We have a manipulator *setw* (a short for set width), it takes as an argument the width in number of spaces. So to print our numbers in four spaces we write `cout << setw(4) << number ;`

Question No: 5 (Marks: 1) - Please choose one

`eof()`, `bad()`, `good()`, `clear()` all are manipulators.

▶ True

▶ False

Stream Manipulations P# 433

www.vuzs.net/

Question No: 6 (Marks: 1) - Please choose one

In functions that return reference, use _____ variables.

▶ Local

▶ Global

▶ Global or static

▶ None of the given option

In functions that return reference, use global or static variables. P# 369

Question No: 7 (Marks: 1) - Please choose one

The declarator of Plus (+) member operator function is

▶ Class-Name operator + (Class-Name rhs)

▶ operator Class-Name + ()

▶ operator Class-Name + (rhs)

▶ Class-Name operator + ()

Page 371,373 example are here

`Complex operator + (Complex &);`

`Complex operator + (parameter-list);`

The syntax of the prototype of the overloaded operator function is:

`return-type operator operator-symbol (parameter-list);`

operator is the keyword here. An example of this will be as follows:

`Complex operator + (Complex &);`

Question No: 8 (Marks: 1) - Please choose one

The compiler does not provide a copy constructor if we do not provide it.

▶ True

▶ False

C will provide the default copy constructor. P#476

www.vuzs.net/

Question No: 9 (Marks: 1) - Please choose one

What is the functionality of the following syntax to delete an array of 5 objects named *arr* allocated using new operator?

delete arr ;

- ▶ Deletes all the objects of array
- ▶ **Deletes one object of array**
- ▶ Do not delete any object
- ▶ Results into syntax error

This statement will call the destructor only for the object pointed by the *arr* and deallocate the space allocated to this object

Question No: 10 (Marks: 1) - Please choose one

What is the sequence of event(s) when allocating memory using new operator?

- ▶ Only block of memory is allocated for objects
- ▶ Only constructor is called for objects
- ▶ **Memory is allocated first before calling constructor**
- ▶ Constructor is called first before allocating memory

For *new* operator, memory block is allocated first before calling the constructor. P#414

Question No: 11 (Marks: 1) - Please choose one

What is the sequence of event(s) when deallocating memory using delete operator?

- ▶ Only block of memory is deallocated for objects
- ▶ Only destructor is called for objects
- ▶ Memory is deallocated first before calling destructor
- ▶ **Destructor is called first before deallocating memory**

For *delete* operator, destructor for the object is called first and then the memory block is deallocated. P# 414

Question No: 12 (Marks: 1) - Please choose one

new and *delete* operators cannot be overloaded as member functions.

- ▶ True
- ▶ **False**

www.vuzs.net/

The overloaded *new* operator returns *void ** when it is overloaded as non-member (global). However, it returns an object pointer like the built-in *new* operator, when overloaded as a member function. P# 414

Question No: 13 (Marks: 1) - Please choose one

The operator function of << and >> operators are always the member function of a class.

- ▶ True
- ▶ **False**

there are two ways of overloading operators, either as class members or non-members. But

these insertion (<<) and extraction (>>) operators cannot be overloaded as members.
P#446

Question No: 14 (Marks: 1) - Please choose one

A template function must have at least ----- generic data type

- ▶ Zero
- ▶ **One**
- ▶ Two
- ▶ Three

We generally use the variable name as *T* (*T* evolves from template). However, it is not something hard and fast. After the variable name, we start writing the function definition. The function arguments must contain at least one generic data type. P# 499

Question No: 15 (Marks: 1) - Please choose one

If we do not mention any *return_value_type* with a function, it will return an _____ value.

- ▶ **int**
- ▶ void
- ▶ double
- ▶ float

The default *return_value_type* is of *int* data type i.e. if we do not mention any *return_value_type* with a function, it will return an *int* value. P# 79

Question No: 16 (Marks: 1) - Please choose one

Suppose a program contains an array declared as `int arr[100]`; what will be the size of array?

- ▶ 0
- ▶ 99
- ▶ **100**
- ▶ 101

Comment: According to me this is true (farhat)

As we know, the array index is one less than the size of the array. P# 103

The size of array is 100. but its index will be from 0 to 99. we will initialize all the elements of the array to 0.

Question No: 17 (Marks: 1) - Please choose one

The name of an array represents address of first location of array element.

- ▶ **True**
- ▶ False

The name of the array is a constant pointer which contains the memory address of first element of the array

Question No: 18 (Marks: 1) - Please choose one

Reusing the variables in program helps to save the memory

▶ **True**

▶ False

Question No: 19 (Marks: 1) - Please choose one

Which of the following option is true about new operator to dynamically allocate memory to an object?

▶ The new operator determines the size of an object

▶ Allocates memory to object and returns pointer of valid type

▶ Creates an object and calls the constructor to initialize the object

▶ **All of the given options**

Question No: 20 (Marks: 1) - Please choose one

new and delete are _____ whereas malloc and free are _____.

▶ Functions, operators

▶ Classes, operators

▶ **Operators, functions**

▶ Operators, classes

Hence, we can call **new** and **delete** operators, P# 342

we have allocated a memory space for our use by **malloc** function. P# 285

Question No: 21 (Marks: 1) - Please choose one

Like member functions, _____ can also access the private data members of a class.

▶ Non-member functions

▶ **Friend functions**

▶ Any function outside class

▶ None of the given options

The friend functions of a class have access to the private data members of class

www.vuzs.net/

Question No: 22 (Marks: 1) - Please choose one

Which of the following statement is best regarding declaration of friend function?

▶ Friend function must be declared after public keyword.

▶ Friend function must be declared after private keyword.

▶ Friend function must be declared at the top within class definition.

▶ **It can be declared anywhere in class as these are not affected by the public and private keywords.**

To declare a friend function, we can put it anywhere in the class. According to the definition of the friend functions, they have access to the private data members of the class.

Question No: 23 (Marks: 1) - Please choose one

The operator function overloaded for an Assignment operator (=) must be

▶ Non-member function of class

▶ **Member function of class**

- ▶ Friend function of class
- ▶ None of the given options

Restrictions on Operator Overloading

The operator overloading functions for overloading (), [], -> or the assignment (=) Operators must be declared as class members.

Question No: 24 (Marks: 1) - Please choose one

For non-member operator function, object on left side of the operator may be

- ▶ Object of operator class
- ▶ Object of different class
- ▶ Built-in data type
- ▶ **All of the given options**

When an operator function is implemented as a non-member function, the left-most operand may be an object of the operator's class, an object of a different class, or a built-in type. Now we discuss it in a detailed manner. (handouts)

Question No: 25 (Marks: 1) - Please choose one

The operator function will be implemented as _____, if obj1 drive the - operator whereas obj2 is passed as arguments to - operator in the statement given below.

obj3 = obj1 - obj2;

▶ **Member function**

- ▶ Non-member function
- ▶ Friend function
- ▶ None of the given options

For member operator, the object on the left side of the + operator is driving this + operation. Therefore, the driving object on the left is available by **this** pointer to + operator function. But the object on the right is passed explicitly to the + operator as an argument.

Question No: 26 (Marks: 1) - Please choose one

Which one of the following is the declaration of overloaded pre-increment operator implemented as member function?

- ▶ Class-name operator +() ;
- ▶ Class-name operator +(int) ;
- ▶ Class-name operator ++() ;
- ▶ Class-name operator ++(int) ;

Question No: 27 (Marks: 1) - Please choose one

The static data members of a class are initialized _____

▶ **At file scope**

- ▶ within class definition
- ▶ within member function

- ▶ within main function

Initialization of static data members is done at file scope which means almost at the vuzs global scope. We initialize it outside of the main.

Question No: 28 (Marks: 1) - Please choose one

Class is a user defined _____.

- ▶ **data type**
- ▶ memory referee
- ▶ value
- ▶ none of the given options.

A class is a user defined data type and it can be used inside other classes in the same way as native data types are used.

Question No: 29 (Marks: 1) - Please choose one

We can also define a user-defines manipulators.

- ▶ **True**
- ▶ False

Parameterized manipulators require one or more arguments. sefill (near the bottom of the iomanip.h header file) is an example of a parameterized manipulator. You can create your own parameterized manipulators and your own simple manipulators.

Question No: 30 (Marks: 1) - Please choose one

Automatic variable are created on _____.

- ▶ Heap
- ▶ Free store
- ▶ static storage
- ▶ **stack**

On the stack, automatic variables are being created and destroyed all the time

Question No: 31 (Marks: 1)

How do we provide the default values of function parameters?

Question No: 32 (Marks: 1)

Why do java consider pointer as dangerous

JAVA, describe pointers as dangerous . if we assign a memory through a pointer where the pointer is destroyed, the memory remains allocated and is wasted. To address these things, there are only references in JAVA instead of pointers. JAVA gives the concept of garbage collection with the use of references. Due to this garbage collection, we are free from the headache of de- allocating the memory. We allocate and use the memory. When it is no longer in use, JAVA automatically deletes (frees) it through garbage collection

Question No: 33 (Marks: 2)

What is memory leak?

suppose, the heap size is decreased as we had allocated memory from it despite the fact that it was never utilized. If this step of allocating memory and then destroy the pointer to this memory carries on then the size of the heap will going on to decrease. It may become of zero size. When there is no memory on heap, the computer will stop running and there may be a system crash. This situation is called a memory leak

Question No: 34 (Marks: 2)

What does optimization the of code means?

Optimization is the process of transforming a piece of code to make more efficient without changing its output or side-effects. The only difference vuzs visible to the code's user should be that it runs faster and/or consumes less memory.

Question No: 35 (Marks: 3)

What is the difference between structure and class?

The ONLY DIFFERENCES between classes and structures are

- 1) classes DEFAULT to having private members. Structures DEFAULT to having public members. These defaults can be changed so classes can be made to work like structures and vice versa.
- 2) classes DEFAULT to inheriting privately from base classes. Structures DEFAULT to inheriting public from base classes. These defaults can be changed so classes can be made to work like structures and vice versa.

Question No: 36 (Marks: 3)

See the following code segment.

```
template <class T>
class myclass {
private:
T x;
public:
myclass (T a) {
x = a;
}
};
```

Write the main function which creates two objects of class for int and double data types.

Question No: 37 (Marks: 3)

Is it possible to define two functions as given below? Justify your answer.

```
func(int x, int y)
func(int &x, int &y)
```

Question No: 38 (Marks: 5)

Write a program using **getline()** member function to inputs a string up to delimiter character comma (,) and then display the string on the screen.

Question No: 39 (Marks: 5)

Do you think that friend functions violate encapsulation? Justify your answer.

Question No: 40 (Marks: 10)

Write a simple program using the **get()** member function of **cin** object reading a text of **30** characters from the keyboard, store them in an array and then using **put()** member function of **cout** object to display them on the screen.

Question No: 41 (Marks: 10)

Write a small program which defines two user-defined manipulators named **octal** and **hexadecimal**. These manipulators should display the decimal numbers into octal and hexadecimal.

In the main function, input a decimal number from the user and then display this decimal number into octal and hexadecimal using user-define manipulators named **octal** and **hexadecimal**.

www.vuzs.net