CS201 SUBJECTIVE QUESTION ANSWERS BY MOHSIN CHINIOTI

03137248280

Q 1: Identify each of the following as system software and application software.
LINUX, DISK CLEANUP, WORD PROCESSOR, WINDOWS, STUDENT INFORMATION

Answer:
System software: - Linux, Disk cleanup, windows.
Application software: - Word Processor, Student information

Q 2: Write a program which defines three variables of type double which store three different values including decimal points, using set precision manipulators to print all these values with different numbers of digits after the decimal number.
Answer:


```
#include
int main ()
{
Double x1 = 12345624.72345
Double x2 = 987654.12345
Double x3 = 1985.23456
cout << setprecision (3) << x1<< endl;
cout << setprecision (4) << x2 << endl;
cout << setprecision (5) << x3<< endl;
Return 0 ;}
```

Q 3: Define static variable also explain life time of static variable?

Answer: Static variable means maintaining the state of a variable. It exists and lives around even when we are outside the function. It is created and initialized only once during the lifetime of the program and therefore it will be destroyed or taken out of memory only once during the lifetime of the program.

Q 4: **What do you know about run time error?**
Answer: Run-Time Errors
• Occur when the program is running and tries to do something that is against the rules. Example: Accessing a non-existent variable, property, method, object, etc (e.g. a method name is misspelled)
• Sources of these can be determined by a careful reading of the code, but unfortunately, not always.

Q 5: **What is limitation of the friendship between classes?**
Answer: Friendship relation between classes is a one way relation that is if one class declare friend another class then the another class is the friend of first class.

Q 6: what is the source and destination of cin?
Answer: For *cin*, the source is normally keyboard and the destination can be an ordinary variable i.e. native-data type Variables.

Q 7: **Write the general syntax of allocation memory dynamically to an array using new operator?**

Answer: Following is the syntax:
New data_type [number_of_locations];

Q 8: What is different between pointers and variable?
Answer: Normal variable contains the value of variable either **int** or **float** whereas pointer variable contains the address of another variables.

Q 9: What is difference between Unary and binary operators and how they can be overloaded?
Answer: Unary operator takes one argument.
a ++ is an example of unary operator
Binary take two operators

+,-,* are example of binary operators
Overloaded binary operator may return any type
Here is general syntax of overloading Return-type operator symbol (parameters);

Q 10: How many types of templates?
Answer: There are two different types of templates in C++ language i.e.' function templates and class templates.

Q 11: What will be the output of following function if we call this function by passing int 5?
Template T reciprocal (T x) {return (1/x) ;}
Answer:
0
The output will zero as 1/5 and its .05 but conversion to int make it zero
Above is prototype of template class so assume passing an int and returning an int.

Q 12: Identify the errors in the following member operator function and also correct them.

```
Math * operator (math m);
Math * operator (math m)
{
Math temp;
temp.number= number * number;
Return number;
```

Answer:

The errors are in the arguments of the member operation function and also in the body of operator member function.

Correct function should be

```
Math *operator (math *m);
Math *operator (math *m)
{
Math temp;
Temp = m;
temp.number= number * number;
Return number;
```


Q 13: Define buffer? Explain its usage?

Answer: A program that writes the output data to the disc, it will be nice to collect the output data (numbers) and write it on the disc in one write operation instead of writing the numbers one by one. The area where we gather the numbers is known as buffer.

Q 14: Why binary search algorithm is efficient than linear search algorithm?

Answer: Binary search algorithm is more efficient than liner algorithm because the arrays are sorted in ascending or descending order and we use "divide and conquer" technique. In binary search each iteration reduces the search but in the linear we have the same number of searches as we have the number of elements. E.g. if we have array of 1000 elements the linear search will take 1000 iterations however binary search will take max 10.

Q 15: Syntax of Post increment and pre increment.
Answer:
Classname operator ++ (); ---- pre increment
Classname operator (int) ++ ---- post increment

Q 16: **What is language translator?**
Answer: we need a translator which translates the code of our program into machine language. There are two kinds of translators which are known as Interpreter and Compilers. These translators translate our program which is written in C-Language into Machine language.

Q 17: **Write something about testing in designing program?**
Answer: Testing. The programmer should design a test plan and use it to test the program. It is a good idea, when possible, to have someone else test the program.

Q 18: Read the given below code and explain what task is being performed by this function.

```
Matrix:: Matrix ( int row , int col )
{
```

```
numRows = row ;
numCols = col ;
Elements = new (double *) [numRows];
For (int i = 0; i < numRows; i ++)
{
Elements [i] = new double [numCols];
For (int j = 0; j < numCols; j ++)
Elements [i] [j] = 0.0;
}
}
```


Hint: This function belongs to a matrix class, having
Number of Rows = numRows
Number of Columns = numCols
Which one (copy constructor or assignment operator) will be called in each of the following code segment?
1) Matrix m1 (m2);
2) Matrix m1, m2;
m1 = m2;
3) Matrix m1 = m2;
Answer:
In this code the matrix function is defined, it gets the number of rows from the user and creates the row of matrix and then gets the columns from the user and creates the columns. The New is showing for creating more array space for the data which user enters. The elements [i] [j] will print the data in matrix.

Q 19: **How many arguments a Unary Operator take? Can we make a binary operator as unary operator?**
Answer: Unary operator takes only one arguments like i++ or i-- (Post increment or post decrement operators for integers) or ++i,--i (Pre increment or pre decrement operators for integers),

we cannot make Unary operator as binary or binary as Unary operator.

Q 20: Which arithmetic operators cannot have a floating point operand?

Answer: Modulus operator

This operator can only be used with integer operands only.

Q 21: What are manipulators? Give one example.

Answer: Manipulators are operators used in C++ for formatting output. The data is manipulated by the programmer's choice of displayed. Endl manipulator: This manipulator has the same functionality as the „\n newline character.

Q 22: Write down piece of code that will declare a matrix of 3x3. And initialize all its locations with 0;

Answer:

```
int matrix [3] [3] ;
Matrix [0] [0] = 0;
Matrix [0] [1] = 0;
Matrix [0] [2] = 0;
Matrix [1] [0] = 0;
Matrix [1] [2] = 0;
Matrix [1] [2] = 0;
Matrix [2] [0] = 0;
Matrix [2] [1] = 0;
Matrix [2] [2] = 0;
```

Q 23: Which one (copy constructor or assignment operator) will be called in each of the following code segment?

1) Matrix m1 (m2);

2) Matrix m1, m2;
m1 = m2;
3) Matrix m1 = m2;
Answer:
1) Matrix m1 (m2); Copy **Constructor**
2) Matrix m1, m2;
m1 = m2; **Assignment Operator**
3) Matrix m1 = m2; **Assignment Operator**

Q 24: What are the advantages and disadvantages of using templates?
Answer:

Advantages:
• Templates are easier to write than writing several versions of your similar code for different types. You create only one generic version of your class.
• Templates are type-safe. This is because the types that templates act upon are known at compile time, so the compiler can perform type checking before errors occur.
• Templates can be easier to understand, since they can provide a straightforward way of abstracting type information.
Disadvantages:
On the other hand, if misused
• Templates can make code difficult to read and follow depending upon coding style.
• They can present seriously confusing syntactical problems. When the code is large and spread over several header files.
• Then, there are times, when templates can "excellently" produce nearly meaningless compiler errors. A common mistake is the angle bracket problem.

Q 25: Suppose a program has a math class having only one data member.
Write the declaration and definition of operator function to overload + operator for the statements of main function.
Math obj1, obj2;
obj2= 10 + obj1;
Answer:

```
#include <iostream.h>
Math
{
mth operator + (obj1,obj2)
mth operator + (obj1,obj2)
{
mth operator + (obj1,obj2)
mth operator + (obj1,obj2)
}
}
```


Q 26: **Is it possible to define two functions as given below? Justify your answer.**
func(int x, int y)
func(int &x, int &y)
Answer: No, we cannot define two functions as func(intx, inty) func(int &x, int&y) because it's give an error function not initializing.

Q 27: **What happens when we use new and delete operator?**
Answer: When we use ***new*** operator to create objects the memory space is allocated for the object and then its constructor is called. Similarly, when we use ***delete*** operator with our objects, the destructor is called for the object before de allocating the storage to the object.

Q 28: **What is the difference between function overloading and operator overloading?**
Answer: Difference b/w function overloading and operator overloading is: In function overloading, the functions have the same name but differ either by the number of arguments or the type of the arguments. Operator overloading is to allow the same operator to be bound to more than one implementation, depending on the types of the operands.

Q 29: **Why the first parameter of operator function for << operator must be passed by reference?**
Answer: Operator<< first parameter must be passed by reference. Its second parameter does not have to be passed as a const-reference parameter; however it is more efficient to pass it by reference than by value.

Q 30: What is the difference between switch statement and if statement.
Answer:
1. If statement is used when we have to check two conditions while switch is a multi conditional control statement.
2. SWITCH statement can be executed with all cases if the "break" statement is not used whereas IF statement has to be true to be executed.

Q 31: How can we initialize data members of contained object at construction time?
Answer: Initialize list is used to initialize the contained objects at the construction time.

Q 32: How the data members of a class are initialized with meaningful values?

Answer: The C++ compiler generates a default constructor for a class if the programmer does not provide it. But the default constructor does not perform any data member's initialization. Therefore, it is good practice that whenever you write a class, use a constructor function to initialize the data members to some meaningful values.

Q 33: Can we overload *new* and *delete* operators?

Answer: Yes, it is possible to overload *new* and *delete* operators to customize memory management. These operators can be overloaded in global (non-member) scope and in class scope as member operators.

Q 34: What will be the output of following functions if we call these functions three times?

(1)

```
Void func1 () {
int x = 0;
X++;
cout << x << endl;
}
```

Answer:

1
1
1

(2)

```
Void func2 () {
Static int x = 0;
X++;
```

cout << x << endl ;
}
Answer:
1
2
3

Q 35: What is the keyword 'this' and what are the uses of 'this' pointer?
Answer: **'this'** is use to refer the current class member without using the name of the class. We cannot use it as a variable name. '**This'** pointer is present in the function, referring to the calling object. This pointer points to the current object.
Q 36: Suppose an object of class A is declared as data member of class B.
(i) The constructor of which class will be called first?
Answer: **(A)**
(ii) The destructor of which class will be called first?
Answer: **(B)**

Q 37: Write the general syntax of a class that has one function as a friend of a class along with definition of friend function.
Answer:
Class frinedclass {
Public:
Friend int compute (exforsys e1)
};
Int compute (exforsys e1)
{
//Friend Function Definition which has access to private data

```
Return int (e1.a+e2.b)-5;
}
```

Q 38: Write a program which defines five variables which store the salaries of five employees, using setw and setfill manipulators to display all these salaries in a column.

Note: Display all data within a particular width and the empty space should be filled with character x

Output should be displayed as given below:

```
xxxxxx1000
xxxxxx1500
xxxxx20000
xxxxx30000
xxxxx60000
```

Answer:

```
#include <iostream.h>
#include <iomanip.h>
Main () {
int sal1 =1000;
int sal2 =1500;
int sal3 =20000;
int sal4 =30000;
int sal5 =60000;
cout << setfill ('x') << setw (10);
cout<< sal1<<endl;
cout << setfill ('x') << setw (10);
cout<< sal2<<endl;
cout << setfill ('x') << setw (10);
cout<< sal3<<endl;
cout << setfill ('x') << setw (10);
cout<< sal4<<endl;
cout << setfill ('x') << setw (10);
```

```
cout<< sal5<<endl;
int i=0;
cin>>i; // to stop the screen to show the output
}
```

Q 39: What is the use of reference data type?
Answer: A reference data type is a variable that can contain an address. The reference data types in Java are arrays, classes and interfaces. However, you could consider a reference data type to be a pointer.

Q 40: What is the "this" pointer? Give an example of its use?
Answer: **This** pointer is use to points to the current object in programming.
In a C++ program, if you create object A of class X, you can then obtain the address of A by using the "this" pointer.

Q 41: If the requested memory is not available in the system then what does calloc/malloc and new operator return?
Answer: malloc returns a void pointer to the allocated space or NULL if there is insufficient memory available. If size is 0, malloc allocates a zero-length item in the heap and returns a valid pointer to that item.

Calloc/malloc and new operator return returns a null pointer to indicate that no memory is available.

Q 42: The following code segment has errors. Locate as many as you can and explain briefly.
Answer:

```
Class Circle // no need to enter colon here, so I removed it
{
Private: //colon missing
```

Double centerX;
Double centerY;
Double radius;
Public: //colon missing
Void setCenter(double, double);
Void setRadius(int);
}; //semi colon missing

Q 43: Write a declaration statement for an array of 10 elements of type float. Include an initialization statement of the first four elements to 1.0, 2.0, 3.0 and 4.0.
Answer: float floatArry [10] = {1.0, 2.0, 3.0, 4.0};

Q 44: **Give the general syntax of class template.**
Answer: template
Class myclass {---};

Q 45: **What is a truth Table?**
Answer:
Sometimes, we find it difficult to evaluate a complicated logical expression. Sometimes the logic becomes extremely complicated so that even writing it as a simple syntax statement in any language.
We know the concept of truth table. The truth tables are very important. These are still a tool available for analyzing logical expressions. We will read logic design in future, which is actually to do with chips and gates. How we put these things together.

Q 46: What will be the output of following code, if user input a number 123?

int input ;
cin >> oct >> input;
cout << hex << input ;
Answer:
53
Rational: it will take 123 as octal and print it in hex form which is 53.

Q 47: What is principle of friendship in the context of functions and classes?
Answer:
Class can declare a friend function and someone from outside the class cannot declare itself friend of a class.
A friend function can access the private variables of class.

Q 48: **Suppose there is a template function 'func' having argument of type U and return type T. What will be the C++ syntax to call this function, passing a variable 'x' of type double and returning an int type?**
Answer:
Template <class T, class U>
T func (T a, U b) {
Return (a<b, a: b);
}
Calling
int i;
Double x;
x = func<int,double> (j,l);

Q 49: Which variable will be used in inner code block if we have the same names of variable at outer code block and inner code block?
Answer: Simply: variable of the inner code is use in the inner code block.

Q 50: **What is the benefit of reference and where can we use it?**
Answer: In references we give the memory address of the object, due to references we pass values without making the copy. Hence, when we have many values & we want efficiency we use references to avoid copy.

Q 51: **Write the C++ code for the declaration of overloaded stream insertion and stream extraction operator for the object "d" of type Date.**
Answer:

```
Date operator >> (date & d1)
{
cout<<d1.day<<"-"<<d1.month<<"-"<<d1.year;
}
```


Q 52: **What is difference between endl and \n?**
Answer:
Endl is manipulator and it inserts new line character and flushes the stream.
\n is control character which is used to insert line break.

Q 53: **What does code optimization mean?**
Answer: It is process by which we make our code such a way that it improves the speed of program. By use of optimization

we refine program codes in this way that it run faster and consume less memory.

Q 54: **How is the following cout statement interpreted by compiler?**
cout << a << b << c ;
Answer:
It will give a compiler error because a, b, c are not declared.

Q 55: **What steps we must follow to design good program?**
Answer:
Hence to design a program properly, we must:
 Analyze a problem statement, typically expressed as a word problem.

 Express its essence, abstractly and with examples.
 Formulate statements and comments in a precise language,
 Evaluate and revise the activities in light of checks and tests.
 Pay attention to detail,
Details: we must check very details of any program.
Reusable: We must write program in such a way that we can reuse them in other program.
Comments: we write the details of important steps in the form of comments.
Code readability: We should use Tab and spaces so codes are easily readable.
User interface: we make interface user friendly.
Q 56: **Suppose we have the following class.**
Class Matrix
{
Private:
int Elements[3][3];

};
Write the operator function of stream extraction operator (>>) for this class.
Answer:
Element operator >> (Element &element){
cout<<element[0][0]<<element[0][1]<<element[0][2];
cout<<element[1][0]<<element[1][1]<<element[1][2];
cout<<element[2][0]<<element[2][1]<<element[2][2];
}

Q 57: Let we have a class,
Class String
{
Private:
Char buf[25];
};
Write code for assignment (=) operator function which assign one String object to other object. Your code should also avoid self assignment.
Answer:
Void String::operator = (const String &other)
{Int length;
Length = other. Length ();
Delete buf;
buf = new char [length + 1];
Strcpy (buf, other.buf) ;}
Q 58: Write down at least the three differences between structure and classes.
Answer:
Structures and classes in C++ are quite similar. The difference between class and structure is of visibility. Every data member

or function written inside the structure is **public** similarly; everything declared inside a class is **private.** While writing classes, good programming practice is to write **private** keyword clearly; similarly, while writing structures, it is good to write the **public** keyword clearly.

Q 59: Which bitwise operator returns true if both bits are different and returns false if both bit are the same?
Answer: Not Operator. (!=)

Q 60: When does a get destroyed? **Page 20**
Answer: When we create an object in a function, this is local to that function. When the function exits the life of the object also comes to end. It means that the object is also destroyed.

Q 61: Why do java consider pointer as dangerous?
Answer:
JAVA, describe pointers as dangerous. 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.

Q 62: What is memory leak?
Answer:
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.
Q 63: What is the difference between structure and class?

Answer: The only difference between classes and structures are classes default to having private members. Structures default to having public members.
In structures, some data variables are gathered, grouped and named as a single entity. A class includes both data members as well as functions to manipulate that data"

Q 64: What are the limitations of the friendship relation between classes?

Answer: Class can declare a friend class from inside and someone from outside the class cannot declare itself friend of a class.

Q 65: How do we provide the default values of function parameters?

Answer: The default value of a parameter is provided inside the function prototype or function definition. For example, we could declare the default function arguments for a function while declaring or defining it.

Q 66: Do you think that friend functions violate encapsulation? Justify your answer.

Answer: The friend functions of a class have access to the private data members of class. We want that our private data is accessible to someone outside, not public for everybody. Otherwise, the data encapsulation and data-hiding concept will be violated. We keep the data members private and declare some

specific functions that are not member of the class but friend of the class. As friends, they have access to the inside data structure of the class despite not being members.

Q 67: What care must be taken while writing Macros?

Answer: Take care of few things while defining macros. There is no space between the macro name and the starting parenthesis. If we put a space there, it will be considered as simple macro without parameters. We can use more than one argument in the macros using comma-separated list. After writing the arguments, enclosing parenthesis is used. There is always a space before starting the definition of the macro.

Q 68: How Avoid Compiler Generated Constructor?

Answer: If a constructor is not defined by the use the compiler generates it automatically. This constructor has no parameter. It does nothing. Although the compiler will create a default constructor for us,

Q 69: What is the function of Flush Command?

Answer: The flush command forces the data from the buffer to go to its destination which is normally a screen or file and make the buffer empty.

Q 70: How to increase the already allocated chunk of memory?

Answer:

Q 71: What does *getline()* member function of cin stream do?

Answer: Another member function of *cin* is *getline()*. It reads a complete buffer i.e. we can write something like:

cin.getline(char *buffer, int buff_size, char delimiter = '\n')

Q 72: When memory is allocated dynamically using new operator within the constructor of class then what is an appropriate place to de-allocate the memory?

Answer: Whenever we allocate memory with the *new* operator, it is our responsibility to de-allocate this memory after the termination of the program. To do this de-allocation, we have an operator *delete*. To de-allocate the memory, allocated with *p = new int;* we will write delete (p);

Q 73: When we call calloc function to allocate memory and its return a NULL pointer what does it mean?

Answer: Calloc function takes two arguments. The first argument is the required space in terms of numbers while the second one is the size of the space. Now we have to see what happens when either we ask for too much memory at a time of non-availability of enough memory on the heap or we ask for memory that is available on the heap, but not available as a single chunk. In this case, the call to calloc will fail. When a call to memory allocation functions fails, it returns a NULL pointer.

Q 74: **What do you mean by garbage collection and how it works in JAVA and C++?**

Answer: 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 But in C and C++ languages, we have to take care of de-allocating the memory. In classes where we use dynamic memory, we have to provide destructors to free this memory.

Q 75: What are the main types of operators in terms of number of arguments they take? **Page 24**

Answer: The difference is in the number of arguments used by the function. In the case of binary operator overloading, when the function is a member function then the number of arguments used by the operator is one. When the function defined for the binary operator overloading is a friend function, then it uses two arguments.

Q 76: Which header file must be included while handling files?

Answer: While handling files, one can have three options. Firstly, we will only read the file i.e. read only file. It means the file is used as input for the program. We need to have a stream for input file i.e. *ifstream* (input file stream). Similarly, if we want to write in some file, *ofstream* (output file stream) can be

used. We can read, write and manipulate the same file using *fstream.h*.

Q 77: What is meant by C++ statement?
Const int *ptr = &x;
Answer: Ptr is pointer to constant integer and (&) is address of "x" to pointer.

Q 78: What is means by default constructor?

Answer: If we do not write a constructor, C++ writes a default constructor for us. The default constructor does not take any parameter. Constructor can take arguments. The default constructor takes no argument. The C++ compiler generates a default constructor for a class if the programmer does not provide it.

Q 79: What is Conversion Constructor?

Answer: A constructor that takes only one parameter is considered a conversion function. We can use a constructor of single argument for conversion operation.

Q 80: Why we keep Utility Function as a Private in Class?

Answer: The functions (methods) are normally written in public part of the class. Are there functions which are private to a class? Answer is yes. Suppose we write a **setDate** function. This function is given an argument and it does the same thing as done by the constructor. In other words, it sets a value of date. Now that function can be public so that it can be called from outside

the class. Now we want that the member functions of the class can call this function. But it should not be called from outside. In this case, we put this function in private section of the class. These functions are called utility functions.

Q 81: What is self Assignment means, why should it be avoided?

Answer: Normally, self -assignment is not directly used in the programs. But sometimes, it is needed. Suppose we have the address of an object in a pointer. We write:

*String s, *sptr;*

Now *sptr* is a pointer to a *String* while *s* is a *String* object. In the code of a program we can write

sptr = &s ; **Page 26**

Self-assignment is very dangerous especially at a time when we have memory manipulation in a class.

Q 82: Is it possible to overload operators for primitive data type? Justify.

Answer: we create the object using the **new** operator. the language has been extended to a programmer to have user defined data types. When we use them in our programs, these are used in the same manner as the primitive data types.

Q 83: C and C++ are free-format languages. What does mean free format?

Answer: C and C++ are free-format languages. We can write the whole code in a single line. But it will be very difficult to read. We format our code so that it could be read easily.

Q 84: What are two types of Conversion for user defined data type?

Answer:

We have a user-defined data type, which we call classes, so we can have pointers to classes.

The functions **tolower** and **toupper** are conversion functions.

Q 85: Describe three important constructs to solve given problems? **Page 27**

Answer: We need simply three constructs to solve any problem.

1) Things should be executed sequentially. That means the statements are executed in a sequence i.e. the second statement follows the first and so on.
2) We need to have a decision. The decision means if something is true, the program executes it. Otherwise, it tries doing something else.
3) The third construct is loop, which is a repetition structure that performs the same task repeatedly with different values.

Q 86: How can we increase the size of dynamic allocated memory in C?

Answer: Whenever, we do not know in advance how much memory space would be required, it is better to use dynamic memory allocation. For example if we want to calculate the average age of students of a class. Instead of declaring an array of large number to allocate static memory, we can ask number of students in the class and can allocate memory dynamically for that number. The C language provides different functions to allocate the memory dynamically.

Q 87: What is alternative of Setw?

Answer: ***setw***, inline manipulator has the alternative function ***cin.width*** and ***cout.width*** with single argument.

Q 88: What is syntax of Switch Statement?

Answer: The syntax of switch statement is as follows.

Switch (variable*/expression)*

{

Case *constant1: statementLlist1;*

Case *constant2: statementLlist2;*

Case *constantN: statementListN;*

Default: *statementList;* }

Q 89: What is difference b/w Pointers and Reference?

Answer:

Pointers

Pointer is a special type of variable that contains a memory address. In C++ language, variables can be without type. So we can have a pointer to an integer, a pointer to a character and a pointer to a float etc.

References

A reference can be considered as a special type of pointer as it also contains memory address. There are some differences between pointers and references. Pointers may point to nothing while references always have to point to something.

Q 90: Write Code for Nested Class?

Answer: Classes defined within other classes are called nested classes. A nested class is written exactly in the same way as a normal class.

```
Class Surround                                    Page 29
{
Public:
 Class FirstWithin
Private:
 Class SecondWithin}
```