XI COMPUTER –MODEL PAPER 2 (2018-19)
WITH SOLUTIONS

Class: XI   **Subject: Computer Science**
Time: 3 Hours   **Max.Marks:70**

Instructions: All Questions are Compulsory.   Programming Language: C++

1. Define Software and its types.   **(2)**
A. Software: Software represents the set of programs that govern the operation of a computer
   System and make the hardware run. Eg: Dos, MS-Word, C++,Visual Basic,Paint Brush, etc.
   Software can be classified broadly into two categories:
   (i)System Software: a)Operating System b) Language Processors (ii) Application Software
2. What are the basic operations of a computer.   **(2)**
A. Basic Operations of a Computer:
   (i) Taking input from the input device and stores it in memory.
   (ii) Processing according to the programs(instructions) and covert input to the output.
       This process will be performed in Microprocessor which consists Arithmetic and Logic
       Unit and Control unit.
   (iii) Displaying output on any output device.
3. Write two types of Micro Computers?   **(1)**
A. Types of Micro Computers:
   i) PDA (Personal Digital Assistants).
   ii) Desktop Computers and Laptop (Notebook) Computers.
   iii) Workstations.
4. What is a Source Code?   **(1)**
A. The program written in a high level language (here in C++) is called as source code.
   (The program that is converted into the machine language is called as object code).
5. Differentiate between the logical errors and runtime errors with one example each?   **(2)**
A. Logical Error: A logical error is that error which causes a program to produce incorrect
   or undesired output. For instance, if we are trying to print the table of a number 3 and if
   we   say   ctr=1;
                      while(ctr>10)
                      {   cout<<n*ctr;
                            ctr=ctr+1;
                      }
 Here the loop would not be executed even once as the condition(ctr>10) is not fulfilled at all.
 Runtime Error (Execution Error):A run time error is that occurs during the execution of a
 program. It is caused of some illegal operation taking place or unavailability of desired or
 required conditions for the execution of the program.
 Eg: Division by zero,
     Trying to open a file which does not exist or it could not be opened,
     If enough memory is not available.
6. Write down the steps to be followed while running a program.   **(2)**
A. (To run a C++ Program, the following steps:
        1. Write the program.
        2. Compile the program.
        3. Run the program.)
     Steps to create a working program:
     (1) Understand the problem well.
     (2) Analyze the problem to
        * Identify minimum number of inputs required for output.
       * Identify processing components.
     (3) Design the program by
        * Deciding step by step solution.

      * Breaking down solution into simple steps.

  (4) Code the program by

    * Identifying arithmetic and logical operations required for solutions.

    * Using appropriate control structures such as conditional or looping control structure.

  (5) Test and Debug your program by

    * Finding errors in it.

    * Rectifying the errors.

  (6) Complete your documentation.

  (7) Maintain your program.

7. What is the use of comments in a program and its types?   **(2)**

A. Use of Comments: **Comments play a very important role as they provide internal documentation of a program.**

With comments, readability and understandability will be increased.

(There are two types of comments depending on number of lines:

  1. Single line comments.   Starts with //

  2. Multi line comments.   Starts with /* and ends with */)

Types of comments:

1. Prologues: The comments in the beginning of a program that summarizes the purpose of the program.

2. Explanatory comments: Wherever applicable, after the statement, keep // and write Comment about that statement.

3. Use comments to help identify {…} pairs greatly and enhance program understanding.

8. What are the characteristics of a good program?   **(2)**

A. Characteristics of a good program:

  (i) Effective and efficient.   (ii) User friendly.

  (iii) Self-Documenting code.   (iv) Reliable.   (v) Portable.

9. An Error in a program is known as _____.   **(1)**

A. Bug

10. Draw the symbol used in the flow chart to read the values?   **(1)**

A.

11. What is the difference between the 'a' and "a"?   **(1)**

A. 'a' is a character literal so it takes 1 Byte to represent.

  "a" is a string literal so it takes 2 Bytes to represent.

12. Write any four escape sequence characters?   **(1)**

 A. (i) \a - Audible bell   (ii)\b - Backspace

  (iii)\n - Newline   (iv) \r – Carriage return

  (v) \t - Horizontal tab   (vi) \0 - Null

13. What will be the result of the following two expressions if i=10 initially?

    1) ++i<=10   2)i++<=10   **(2)**

A. (i) False   (ii) True

14. Explain tokens in details with example.   **(3)**

A. Token: The smallest individual unit in a program is known as a token (Lexical Unit).

  There are 5 types of tokens.

  (i) Keywords (ii) Identifiers (iii) Literals (iv) Punctuators (v) Operators.

  (i) Keywords: Keywords are the words that convey a special meaning to the language compiler. These are reserved for special purpose and must not be used as normal identifier names. Original C++ consists 48 keywords and ANSI C++ consists 63 keywords.   Eg: int, switch, while, etc.

  (ii)Identifiers:Identifiers are the names given variables, objects, classes, functions, arrays, etc. Eg: rno, myfile, etc.

  (iii)Literals (Constants):are data items that never change their value during a program run.

Different types of literals...

(i) Integer constant   (ii) Floating Constant (iii) Character Constant
(iv) String Constant.   (v) boolean literal

(iv) Operators :The operations (specific tasks) are represented by operators and the objects of the operation(s) are referred to as operands.

They can be classified into 3 categories depending on number of operands it is working:

(a) Unary operators: Works on single operand. Eg: ++ , -- , sizeof, etc
(b) Binary operators: Works on two operands. Eg: +, - , *, /, %, <, >, etc.
(c) Ternary operator: Works on three operands. Eg: Conditional operators.
(exp1)?exp2:exp3

(v) Punctuators (Separators): [,], (,), { , } , ;, :, #, etc.

15. Find errors if any in the following statements.   **(2)**

```
cout<<"X=";X;
cin>>x;>>y;
cout<<"Enter a number";
cin>>x;
```

A. **Errors in Line 1:**   (a) ; is given in place of <<   (b)X is given in place of x

cout<<"X=**;**X ;   // Correct: cout<<"X="<<x;

Error in Line 2:   ; is given in the middle of the line

cin>>x**;**>>y;   //Correct: cin>>x>>y;

16. Define operators and its types?   **(4)**

A. **Operators :**The operations (specific tasks) are represented by operators and the objects of the operation(s) are referred to as operands.

They can be classified into 3 categories depending on number of operands it is working:

(a) Unary operators: Works on single operand.

Eg (i) ++ (Increment operator). Used to increment 'one' to the existing value.
A=10;   A++;
After the execution of above statement, A value will becomes as 11.

(ii) - - (Decrement Operator). Used to decrement 'one' to the existing value.

(iii) sizeof( ): This is used to return the number of bytes for the given data type or variable.
int a;
sizeof(a);   //returns 2 since a is of type int.

(iv) Unary + : It will give the value with same sign.
A=10;   +A will gives 10 only.
A= -10;   +A will gives -10.

(v) Unary - : It will give the value with opposite sign.
A=10;   -A will gives – 10.
A=-10;   -A will gives 10.

(vi) !: Not

(b) Binary operators: Works on two operands.

Eg: Arithmetic Operators: + , - , * , / , %
+ for addition on two operands.   - for subtraction
* for multiplication   / for quotient
% for remainder.
Relational Operators: < , <=, >, >=, !=, = =.
Logical Operators: &&, ||
Etc.,

(c)Ternary operator: Works on three operands.
Eg: Conditional operators. (exp1)?exp2:exp3
Here when the expression 1 becomes true expression 2 will be executed and if it becomes false, then expression 3 will be executed.

17. void main( )   **(2)**
   { int val,res,n=1000;
     cin>>val;
     res=n+val >1750 ?400:200; //Note:Actually instead of >, ? was printed, I corrected it.
     cout<<res; }
       1. If val=750; the res=?
       A. 200
       2. Name of the operator used in the program?
       A. Conditional Operator

18. Generate the series 0,1,1,2...N   **(4)**
   //Program to generate fibonacci numbers.
   #include<iostream.h>
   #include<conio.h>
   void main( )
   {clrscr( );
     int first,second,third,i,n;
     cout<<"\nHow many numbers you want to generate the fibonacci numbers..";
     cin>>n;
     first=0;
     second=1;
     cout<<endl<<first<<"\t"<<second<<"\t";
     for(i=3;i<=n;i++)
     { third=first+second;
       cout<<third<<"\t";
       first=second;
       second=third;
     }
     getch( );
   }

19. What are Iteration Statements? Explain them in detail.   **(6)**
  A. The iteration statements allow a set of instructions to be performed repeatedly until a
     certain condition is fulfilled. The iteration statements are also called loops or looping
     statements. C++ Provides three kings of loops. (i) for (ii) while (iii)do-while loop.
             All three loop constructs of C++ repeat a set of statements as long as a specified
     condition remains true. This specified condition is generally referred to as a loop control.
     For all three loop statements, a true condition is any nonzero value.
     A zero value indicates a false condition.
     <u>Parts of a loop:</u>
       (i) Initialization expression(s)   (ii)Test expression.
       (iii)Update expression(s)   (iv)Body of the loop.
     (i) For loop:
         Syntax: for (initialization expression(s); test expression; update expression(s))
                       body of the loop;
         Firstly, initialization expression is executed. Then the test expression is evaluated,
     which results into either true(1) or false(0). If the test expression is true, then the body of the loop
executed otherwise control come out of the for loop. After execution of the body of the loop, then the
update expression will takes place. Then condition will be tested, ...the process will repeat until
the condition evaluates to false.
         Eg: for (i=1;i<=10;i++)
                   cout<<i;
(ii) While loop (Entry controlled loop):
   Syntax: while (test expression)
                     body of the loop;

The loop variable must be initialized before enter into the while loop and it should be updated inside the boy of the while loop.

The condition will be tested before entering into the loop. So if initially the condition evaluates to false, then the loop will not execute at least once.

     Eg: i=1;
        while (i<=10)
       {   cout<<i;
        i++;
       }

(ii) Do- While loop (Exit controlled loop):
     Syntax: do
           {body of the loop;
            } while (test expression)
              body of the loop;

The condition will be tested after execution of the loop for once. So if initially the condition evaluates to false, then the loop will execute at least once.(since the check will be performed at the end)

     eg: i=1;
       do
       {   cout<<i;
        i++;
       }while (i<=10);

20. Write a program to transpose the matrix of size M*N?   **(3)**

```cpp
//program to find transpose of M*N Matrix
#include<iostream.h>
#include<conio.h>
void main( )
{ int A[10][10],B[10][10];
  int M,N,i,j;
  cout<<"\nEnter the number of Rows: ";
  cin>>M;
  cout<<"\nEnter the number of Columns: ";
  cin>>N;
  cout<<"\nEnter "<<M*N<<" elements...\n";
  for(i=0;i<M;i++)
    for(j=0;j<N;j++)
    {   cin>>A[i][j];
        B[j][i]=A[i][j];
    }
  cout<<"\nThe entered Array..........\n";
  for(i=0;i<M;i++)
  { for(j=0;j<N;j++)
        cout<<A[i][j]<<"\t";
    cout<<endl;
  }
  cout<<"\nThe transpose of the matrix.......\n";
  for(i=0;i<N;i++)
  { for(j=0;j<M;j++)
        cout<<B[i][j]<<"\t";
    cout<<endl;
  }
  getch( );
}
```

21. Explain various types of Constants with example. **(3)**

A. **Constants (Literals): are data items that never change their value during a program run.**
Different types of literals…
(i) Integer constant   (ii) Floating Constant (iii) Character Constant (iv) String Constant.
(i) Integer Constant: Integer constants are whole numbers without any fractional part.
Three types of integer constants.
(a)Decimal integer constants (b) Octal Integer Constants (c) Hexadecimal integer Constants
Eg: 789 – Decimal integer constant
032 – Octal integer constant
0xFA2 – Hexadecimal integer constant.
(ii) Floating Constants (Real Constants): Real constants are numbers having fractional
Form. These may written in fractional form or exponential form.
Eg: 17.45   - Fractional form
0.1745E2 – Exponential form
(iii)Character Constants: It is a single character kept in between pair of ' '.
Any ASCII Character can be represented as a character constant.
An escape sequence, which consists non-graphic characters can be represented through a
character constant.
Eg: 'n', '\n', 'z','\a', '\t', 'k', etc.
(iv) String Constants: Multiple character constants are treated as string literals. A string
Literal is a sequence of characters surrounded by double quotes. Each string literal is by
default added with a special character '\0'.
Eg: "abc"   - size is 4.   "Navodaya" – size is 9.

22. Write a program to compare two strings without using functions. **(3)**

A. `//Program to compare two strings.`

```
#include<iostream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
void main()
{char str1[10],str2[10];
 int l1=0,l2=0,i,flag=0;
 clrscr( );
 cout<<"\nEnter the first string: ";
 gets(str1);
 cout<<"\nEnter the second string: ";
 gets(str2);
 for(i=0;str1[i]!='\0';i++);
 l1=i;
 for(i=0;str2[i]!='\0';i++);
 l2=i;
 if(l1!=l2)
 {   cout<<"\nBoth strings are not the same...";
    getch( );
    exit(0);
 }
 for(i=0;str1[i]!='\0';i++)
 {   if(str1[i]!=str2[i])
    { flag=1;
        break; }
 }
 if(flag= =0)
    cout<<"\nEntered strings are identicle...";
```

```
        else
          cout<<"\nEntered strings are not identicle...";
        getch( );
      }
```
23. Write a program to find the factorial of 10 using recursion?   **(4)**

A.
```
    #include<iostream.h>
    #include<conio.h>
    long f=1;
    long factorial(int n)
    {if (n==0)
        return f;
     else
        f=n*factorial(n-1);
    }
    void main( )
    { clrscr( );
      cout<<"\nThe factorial of the number 10 = "<<factorial(10);
      getch( );
    }
```
24. Explain Storage classes in detail?   **(3)**

A. Storage Class Specifiers: The storage class specifiers tell the compiler how to store the subsequent variable(s). C++ provides four storage class specifiers.

(a) auto   (b) register (c) static   (d) extern

Syntax:

  Storage-specifier type variable-name;

(a) auto: It refers that it is automatic variable. By default, the variables defined within a function are automatic. An automatic variable is not created until the function, in which it is defined, is called. The auto variable is automatically created when the function is called and automatically destroyed when it terminates. An auto variable is alive as long as the function is executing. Thus, the lifetime of the variable is the time during which its parent function is running. The scope of an auto variable is the function scope. They can be accessed only from their parent function.

(b) register: A register declaration is an auto declaration. A register variable has all the characteristics of an auto variable. The only difference between the two is that register variables provide fast access as they are stored inside CPU registers rather than in memory. So, the time required to read from memory is saved. The register as well as auto can be applied only to local variables.

(c) extern: A C++ program may split in different files. In such case, if you have declared a global variable in a file, C++ does not allow you to declare it again in some other file. If you keep extern specifier before variable(s) declaration, tells the compiler that the variable types and names that follow it have been declared elsewhere, so that, fresh memory is not allocated to these variables. The extern specifier is only for global variables. The life time of external variables is the life of the program. The scope of external variables is the file scope.

(d) static: There can be static global variables as well as static local variables. The static variables are permanent within their own function or file. They are not known outside their function or file but they maintain their values between calls. A normal global variable has program scope, a static global variable has file scope.

 When static modifier applies to a local variable, it is initialized only when the very first call to the function that occurs. It is not destroyed when the function terminates, rather it holds its value even after function's termination but it can be only accessed within its own functioin.

25. **Write the output of the following program.** (3)

```
#include<iostream.h>
int func(int &x,int y=10)
{   if (x%y= = 0)  return ++x;
        Else return y- -; }
 void main( )
{ int p=20,q=23;
  q=func(p,q); cout<<p<<q<<endl;
  p=func(q);cout<<p<<q<<endl;
  q=func(p); cout<<p<<q<<endl; }
```

A. 2023
    1023
    1111

26. **Explain Accumulator and Program Counter.** (3)

A. **Accumulator (AC):** The accumulator register is a general purpose processing register. It is a buffer register that stores intermediate results during a computer run.
Program Counter(PC): The program counter sends to the memory the address of the next instruction to be fetched and executed.

27. **Write short note on Hard Disk..** (2)

A. Hard disk is a secondary memory device. The hard disk memories store information on one or more circular platters(disks) which are continually spinning. These rotating disks are coated with a magnetic material and stacked with space between them. Information is recorded on the surface of rotating disks by magnetic heads as tiny magnetic spots. These heads are mounted on access arms. Information is recorded in bands. Each bank of information on a given disk is called a track.
        The tracks are commonly divided into pie-shaped sections called sectors. In most systems, the minimum quantity of information which can be transferred is a sector.   A motor rotates the disk at a rapid speed. Data are recorded on the tracks of a spinning disk surface and read from the surface by one or more read/write heads.The hard disks of today have storage capacity measured in giga bytes, most common being 160 and 250 GB.
(Concentric circles on the magnetized surface of the magnetic disks known as Tracks. The tracks on the disk surface are divided into invisible segments known as Sectors.)

28. **Expand ASCII, ISCII.** (1)

A. **ASCII** – American Standard Code for Information Interchange.
   ISCII - Indian Standard Code for Information Interchange.

29. **Base of Octal and Hexadecimal is ___ and ___.** (1)

A. **Base of Octal = 8**
   Base of Hexadecimal = 16

30. **Why do Computers use Binary Number System?** (3)

A. In our real life we use decimal number system, which consists totally 10 digits. It is very difficult to design electronic equipment so that it can work with 10 different voltage levels On the other hand, it is very easy to design simple, accurate electronic circuits that operate with only two voltage levels. For this reason, almost every digital system uses the binary number system (base 2) as the basic number system of its operations, although other systems are often used in conjunction with binary.
        In the binary system there are only two symbols or possible digit values, 0 and 1. Even so, this base-2 system can be used to represent any quantity that can be represented in decimal or other number systems.
        So Computers use Binary Number System.