| U.S.N. | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|

# B.M.S. College of Engineering, Bengaluru-560019

**Autonomous Institute Affiliated to VTU**

## July 2024 Semester End Main Examinations

**Programme: B.E.** **Semester: V**
**Branch: Electronics and Communication Engineering** **Duration: 3 hrs.**
**Course Code: 22EC5PE1OP** **Max Marks: 100**
**Course: Object Oriented Programming using C++**

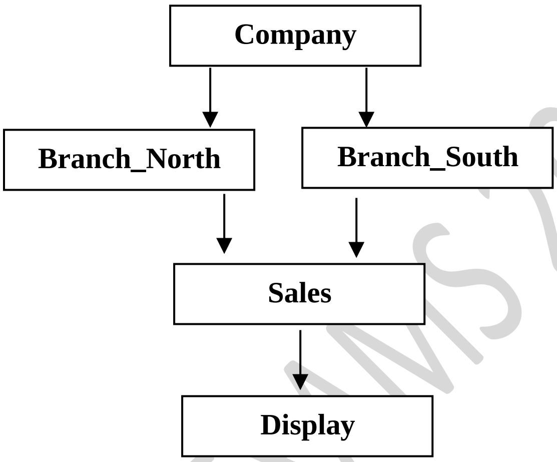
**Instructions**: 1. Answer any FIVE full questions, choosing one full question from each unit.
2. Missing data, if any, may be suitably assumed. Syntax errors are not intentional.
3. ***For every code meaningful comments and sample output MUST be given.***

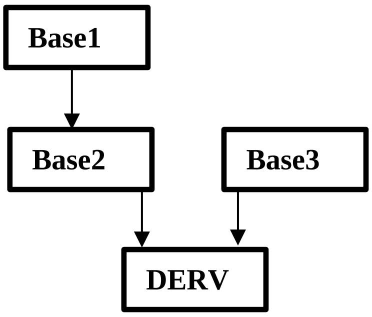
**Important Note:** Completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages. Revealing of identification, appeal to evaluator will be treated as malpractice.

<table>
<tr><th></th><th></th><th>UNIT - I</th><th><i>CO</i></th><th><i>PO</i></th><th>Marks</th></tr>
<tr><td>1</td><td>a)</td><td>Create a structure (STUDENT) to hold student details. Write functions store(STUDENT) and display(STUDENT) to facilitate storing and displaying details of a student. Generate roll numbers automatically.</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>09</b></td></tr>
<tr><td></td><td>b)</td><td>Create an enum to hold the primary colours (R,G,B) and outline how data can be stored and retrieved from it.</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>05</b></td></tr>
<tr><td></td><td>c)</td><td>In an application, a function and main program is written as depicted below:
<pre>
sample(int p, float q, int a=10, float
b=5.9, char ch=’ ‘)
{ cout<<p<<endl<<q<< endl<<a<< endl<<b<<
endl<<ch; }

main()
{
int n1 = 11; int n2=12;   int n3 = 13;
float m1 = 1.1;  float m2 = 2.2;
char c1 = ‘A’;
sample(n1,m1,n2);
sample(n2,m2,n2,c1);
sample(n1,m1);
sample(n1,n2,n1,2.0);
sample(100,50.0,’Q’);
}
</pre>
Write the output with proper justification.</td><td><i>CO2</i></td><td><i>PO3</i></td><td><b>06</b></td></tr>
</table>

|  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|
|  |  |  | **UNIT - II** |  |  |  |
|  | 2 | a) | Create a class (OOP2) to hold one integer, one decimal and one char variable. Include functions for the following program to compile:<br>`void main()`<br>`{   OOP2 obj1, obj2;`<br>`Add(obj1,obj2);   // to add the int and float data and PRINT the result, do not store the result`<br>`Add(obj1);       // to double the value of obj1`<br>`obj1.Add(obj2);  // to add the data of obj1 to obj2 and keep the result in obj1`<br>`}`<br>What OOP concepts do you encounter in this above example? | *CO2* | *PO3* | **10** |
|  |  | b) | With a class called EXAMPLE with suitable members outline the concept of constructors (all types). Illustrate execution with a main function. | *CO1* | *PO1* | **10** |
|  |  |  | **UNIT III** |  |  |  |
|  | 3 | a) | Create a class 'complex' to hold a complex number of the form x + iy. Store the real and imaginary part as data members. Include functions for the following code to compile:<br><br>void main()<br>{<br>complex c1,c2,c3;<br>c1 = c2+c3;<br>c1+=c2;<br>c1++;<br>cout<<c2++; //display the integer value only<br>} | *CO2* | *PO3* | **12** |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | b) | In a certain company database is maintained as follows: The class company consists of Company name, No_of_branches, Address, licence_number. License_number is NOT to be shared. The purpose of this class is to JUST serve as a base class to other classes, as depicted in the diagram below:<br>The classes: ‘Branch_North’ and ‘Branch_South’ contain emp_ID, emp_name as private data, and display function as sharable data. They also maintain data about no of products sold, revenue generated as non-sharable data. Display class has no data of its own and will display data of the 2 branch classes and company name and address.<br>Develop classes and list the features you encounter in this example.<br> | *CO2* | *PO3* | **08** |
| | | | **OR** | | | |
| | 4 | a) | The following statement is to be compiled, where data1, data2, data3 are all objects of a class’EXPRSON’. Write functions for the same. Assume suitable data type for the data of the class.<br>if (data1 < data2)<br>        data1+=5;<br>else<br>data1 = data2 *data3++; | *CO2* | *PO3* | **10** |

<table>
<tr><td></td><td></td><td>b)</td><td>Realize classes as per the given specification: All classes have private, protected and public data. All classes have a function display(). Specify how run time ambiguity is resolved when display function is to be invoked. Specify the contents of an object of class DERV. Consider protected mode of inheritance in all places.<br><br> </td><td><i>CO2</i></td><td><i>PO3</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td></td><td><b>UNIT - IV</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>5</td><td>a)</td><td>Develop a class to store 50 data items of any data type. Write store and retrieve functions.</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>In an application to create password for access, 3 different types of exception may occur. One when the first character of password is a number, one if any character is a ‘.’ and one when the number of characters are less than 8. All these are to be dealt uniquely. Create such an application using appropriate OOP concepts (those pertaining to syllabus ONLY)</td><td><i>CO2</i></td><td><i>PO3</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td></td><td><b>UNIT V</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>6</td><td>a)</td><td>Write codes for displaying the following using appropriate stream functions.
<table>
<tr><td><b>1</b></td><td><b>0</b></td><td><b>1</b></td><td><b>3</b></td><td></td><td></td><td></td></tr>
<tr><td><b>5</b></td><td><b>8</b></td><td><b>0</b></td><td></td><td></td><td></td><td></td></tr>
<tr><td><b>#</b></td><td><b>#</b></td><td><b>#</b></td><td><b>1</b></td><td><b>.</b></td><td><b>0</b></td><td><b>5</b></td></tr>
<tr><td><b>#</b></td><td><b>#</b></td><td><b>1</b></td><td><b>2</b></td><td><b>.</b></td><td><b>0</b></td><td><b>0</b></td></tr>
</table></td><td><i>CO2</i></td><td><i>PO3</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Write the data ‘Object Oriented Programming’ onto a text file “BMS’. Read every 5th data from the file and store in a variable. Display the contents of the variable.</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td></td><td><b>OR</b></td><td></td><td></td><td></td></tr>
<tr><td></td><td>7</td><td>a)</td><td>An inventory system requires output to be printed in the following format. Facilitate the same using appropriate iostream functions ONLY. Do not use simple cout statements.
<table>
<tr><td>ITEM</td><td>COST</td><td>No_Units_Available</td></tr>
<tr><td>AABBSS</td><td>0100.00</td><td>1000</td></tr>
<tr><td>ABA</td><td>**10.00</td><td>18000</td></tr>
<tr><td>SQUR</td><td>*128.50</td><td>50</td></tr>
</table></td><td><i>CO2</i></td><td><i>PO3</i></td><td><b>10</b></td></tr>
<tr><td></td><td></td><td>b)</td><td>Using a single file pointer demonstrate how we can work with 2 files.</td><td><i>CO1</i></td><td><i>PO1</i></td><td><b>10</b></td></tr>
</table>

******