

## OPEN ELECTIVE

BE 3<sup>RD</sup> AND 4<sup>TH</sup> YEAR, 2<sup>ND</sup> SEMESTER EXAMINATION, 2023

## OBJECT ORIENTED PROGRAMMING FUNDAMENTALS

Time: 3 hours

Full Marks: 100

Answer all parts of a question together in one place. Do not scatter the answers.

<p>CO1 [25 MARKS]</p>	<p>1. a) Justify the truth/falsity of each of the following statements. Provide supporting arguments and/or code snippets (where needed) for justification. (Any 3)</p> <p>i) Default values in function parameters must be supplied from left to right.  ii) We can make a function inline by just specifying the keyword <i>inline</i> before the function signature.  iii) Call by reference takes lesser space than call by value.  iv) Function call statement must always appear on the right hand side of assignment operator.</p> <p>b) Distinguish among each of the following triples. (Any 3)</p> <p>i) Iteration vs. normal recursion vs. tail recursion  ii) Array vs. structure vs. class  iii) malloc() vs. calloc() vs. new  iv) Local variables vs. reference variables vs. pointer variables</p> <p>c) Write a recursive function to calculate <i>a to the power b</i> (<math>a^b</math>). Both a and b are taken from user. Hence, calculate the sum of the following series:  <math>1+8+81+1024+\dots\dots\dots n</math> terms. (User will decide the value of n at runtime)</p> <p style="text-align: center;">Or,</p> <p>Create an array of N number of elements dynamically. Next pass it to a function that finds out the highest element from the array recursively. N should be taken from user at runtime.</p> <p style="text-align: right;">[(3x3)+(3x3)+7=25]</p>
<p>CO2 [30 MARKS]</p>	<p>2. a) Fill up the blanks with suitable phrases. Hence justify the validity of each of the completed statements with supporting reasons. Provide code snippets where needed.</p> <p>i) Data encapsulation means _____ and it can be achieved by _____.</p> <p>ii) Two functions cannot be overloaded on the basis of _____ only. Their _____ must differ.</p> <p>iii) <i>this</i> operator cannot be used within _____ because _____.</p> <p>iv) The compiler always provides one _____ for a class only if we _____.</p> <p>v) Only the _____ member variables of a class can be accessed within _____ functions.</p> <p>vi) Copy constructor always takes _____ as parameter because _____.</p> <p>vii) A constant object of a class can be used to invoke only the _____ methods so that _____.</p> <p>viii) Constructors can be _____ because _____. However the _____ cannot be.</p> <p>ix) Only the _____ member variables of a class can be changed within _____ functions.</p> <p>x) Usage of friend classes/functions is _____ because _____.</p> <p>xi) Function overloading is also known as _____ time polymorphism or _____ binding.</p> <p>xii) A constant member variable of a class can be initialized by _____.</p>

[Turn Over]

	<p>b) Assume a class <i>Money</i> as the following definition. Now complete the class definition in order to properly execute the statements in the <i>main()</i> method. Finally discuss the output.</p> <pre> class Money {     int rupees, paise; };  int main() {     Money m1(450,90), m2(500,70),m4;     Money m3=m1;     m3.add(m2);     m3.show();     m4=subtract(m2,m1);     m4.show(); } </pre> <p>Or,</p> <p>Consider the following code snippet. Initialize the variable <i>s</i> with 10 and hence discuss the output of the code with proper reasons.</p> <pre> class Sample {     int a;     static int s; public:     Sample (int x=0)     { s++;       a=s+(++x);     }     void show()     { cout&lt;&lt;"a="&lt;&lt;a&lt;&lt;"s="&lt;&lt;s&lt;&lt;endl;     } };  int main() {     Sample s1(3), s2;     s1.show();     s2.show();     Sample ob[3];     for(int i=0;i&lt;3;i++)         ob[i].show();     Sample s3(i+1);     s3.show();     Sample s4(i+2);     s4.show(); } </pre> <p style="text-align: right;">[(2x12)+6=30]</p>
<p>CO3 [30 MARKS]</p>	<p>3. a) Consider the following class <i>Sample</i>. Update its definition to perform the tasks as specified in <i>main()</i>. Clearly indicate which portion of the class is dedicated for which task. Finally discuss the output.</p> <pre> class Sample {     int s; };  int main() { Sample ob1(3), ob2(4), ob3;   ob1+=2;   ob3=ob1*(ob2+5);   cout&lt;&lt;ob3;   ob2=20 + ob3++;   cout&lt;&lt;ob2&lt;&lt;endl&lt;&lt;ob3; } </pre> <p>Or,</p> <p>Distinguish between each of the following pairs: Provide suitable code snippets in support of your answer.</p> <ol style="list-style-type: none"> <li>i) Normal virtual function vs. pure virtual function</li> <li>ii) Normal destructors vs. virtual destructors</li> <li>iii) Late binding vs. early binding</li> <li>iv) Multiple inheritance vs. hybrid inheritance</li> </ol>

	<p>b) Consider the following two classes <i>A</i> and <i>B</i>. Complete their definitions to execute the statements specified in <i>main()</i>. Clearly indicate which portion of the class is dedicated for which task. Finally discuss the output. <b>The order of the classes should not be changed.</b></p> <pre> class A { int a; };  class B { int b; };  int main() { A oba1;   B obb1 (5);   oba1=obb1;   oba1-&gt;show();   B obb2,obb3;   obb3=obb2(10);   cout&lt;&lt;obb2&lt;&lt;endl&lt;&lt;obb3;   obb3-==+obb1;   cout&lt;&lt;obb1&lt;&lt;endl&lt;&lt;obb3; } </pre> <p>c) Fill up the blanks with appropriate phrases. Hence justify the validity of each of the completed sentences. (Use code snippets if/where required for justification)</p> <p>i) A pure virtual function, even if _____ the class makes it _____.</p> <p>ii) The _____ of a class _____ the class. Otherwise, _____ error is reported.</p> <p>iii) For overloading input and output operators, we always have to use _____ functions because _____.</p> <p>iv) For converting a basic data type to an object of a class, we need _____. However, for the reverse we need _____.</p> <p>v) The _____ data members of a class have more strict accessibility than public data members, but less strict accessibility than _____ data members.</p> <p>vi) For [ ] operator overloaded function, it is better to return _____ because _____.</p> <p style="text-align: right;">[8+10+(2x6)=30]</p>
<p><b>CO4</b></p> <p><b>[15 MARKS]</b></p>	<p>4. a) Do as directed:</p> <p>i) <i>If an object a class is created before a try block, then on coming out of that block, the destructor of the class is called first and the catch block is executed next.</i>—(Justify the truth/falsity of the statement with valid reason.)</p> <p>ii) There is a provision of rethrowing an exception if _____. (Fill in the blank with appropriate phrases. Use code snippet in support of your answer.)</p> <p>iii) <i>After executing the statements in the catch block, the program control again returns to the try block.</i> (State true or false. Provide reasons and/or code snippets in support of your answer)</p> <p>iv) If we are not sure about the exact type of exception being thrown, then how can we handle such scenario? (Show with suitable code snippet)</p> <p>b) What are the possible modes in which a file can be opened? State the significance of each of them.</p> <p>c) Show how a custom exception class object can be created, thrown and handled. (Use code snippet to show)</p> <p style="text-align: right;">[(2x4)+3+4=15]</p>

**Course Outcomes:**

**CO1: Recognize and illustrate** the basic concepts of OOP and its enhancements over procedural languages.

**CO2: Explain, illustrate and recognize** the basic features of classes and objects.

**CO3: Explain and demonstrate** the extended features of OOP (Inheritance, Polymorphism, Operator overloading).

**CO4: Illustrate** basic I/O, file I/O and exception handling.