

Military Institute of Science and Technology (MIST)

Department of Computer Science & Engineering (CSE)

CSE 205: Object Oriented Programming Language, CT-3, Date: 23/06/21

Time: 40 min	Marks: 20
Instructions: <ol style="list-style-type: none"> Making choices in life is pretty hard, fortunately, you don't have to make any in this exam. Answer all the questions provided below. Numbers on the right side of each question signify individual question marks. 	
1.	<div data-bbox="1435 612 1507 666">3+3</div> <div data-bbox="248 634 1320 1276"> <pre> classDiagram class Base { f1()=0 f2()=0 } class Derived1 { f1(){...} } class Derived2 { f2(){...} } class Derived3 { f3()=0 f4()=0 } class Derived4 { f3(){...}, f4(){...} } Base < -- Derived1 Base < -- Derived3 Derived1 < -- Derived2 Derived3 < -- Derived4 </pre> </div> <p>Take a look at the inheritance scenario above. Now, based on it, answer the following questions-</p> <ol style="list-style-type: none"> Identify Abstract Class(es), Interface(s) and Concrete Class(es). Which ones are Pure Virtual Functions in this scenario? Why do you think we keep these functions in base classes even though they aren't used by the base class itself?
2.	<div data-bbox="1435 1478 1507 1532">6+2</div> <div data-bbox="162 1499 1391 2075"> <pre> classDiagram class Shops { int shop_ID String Owner_Name Shops(...) constructor } class Small_Shops { double reg_num int num_of_employees Small_Shops(...) } class Mega_Shops { Mega_Shops(...) } class Big_Shops { double tax_ID int num_of_employees int num_of_managers Big_Shops(...) } Shops < -- Small_Shops Shops < -- Mega_Shops Small_Shops < -- Big_Shops Mega_Shops < -- Big_Shops </pre> </div> <p>Identify the problem the above inheritance scenario poses. Explain the drawbacks of this situation with a graphical example or a short code snippet.</p> <p>Can virtual inheritance solve this problem? If so, which class needs to be inherited virtually?</p>
3.	<div data-bbox="1435 2352 1507 2440">3x2 =6</div>

```

4  class complex_num
5  {
6  private:
7      int real;
8      int img;
9  public:
10     complex_num(int r, int i)
11     {
12         real = r;
13         img = i;
14     }
15 };
16 int main()
17 {
18     complex_num C1(5,10); /// C1 = 5 + 10i
19     complex_num C2(10,-5); /// C2 = 10 - 5i
20
21     C1 += C2; ///C1 = 15 + 5i
22
23     C2 = 10 + C2; ///C2 = 20 + 5i
24
25     int a = C2; ///a = 20;
26     ///(only C2's real part is stored in a)
27
28 }

```

Take a look at the code snippet above. Now, do necessary operator overloading so that lines 21, 23, and 25 can be executed. **Write only the operator overloading functions for the 3 cases.**

Note that, “complex_num” it’s a class that stores complex numbers. The value of each object/variable after the line is executed is given as comments in the corresponding lines.