SASTRA

ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION

DEEMED TO BE UNIVERSITY

(U/S 3 OF THE UGC ACT, 1956)

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

School of Computing

End Semester Examination – July 2021

Course Code: CSE201 – (Semi-Lab)
Course Name: Object Oriented Programming in C++

Duration: 120 minutes
Max Marks: 50
Number of Questions: 02

Instructions

- You will be given two questions with 25 marks each
- Execute the programs and prepare a single PDF file containing both the programs and its output
    - Create a word file.
    - Type your Name and register number.
    - Copy the programs and paste it in the word file (Screenshots of the code will not be considered for evaluation)
    - Take a snapshot of the output & paste it after the corresponding program.
    - Convert the word file to pdf file, then upload
- Name your file as follows: 12010001_Lab.pdf (Give your reg.no)
- Submissions with multiple files will not be evaluated.
- Exam duration is between 9:30 am to 11:30 am
- Upload time: 11:30 am to 11:45 am

Question 1 (25 Marks)

Develop a C++ program to overload the arithmetic operators + and the relative operator >= in the following class. The class represents computer memory size in terms of GB, MB, KB and bytes. Assume that:

1 KB =1024 bytes
1 MB = 1024 KB
1 GB =1024 MB

Overload + operator such that when it used as object3 = objec1 + object2 then it should carry out the sum of the two objects as follows:

Bytes = bytes of object1 + bytes of object2
KB = KB of object1 + KB of object2
MB = MB of object1 + MB of object2
GB = GB of object1 + GB of object2

The resultant values should be inserted into a nameless object and returned. While adding the bytes of the two objects if the sum exceeds 1024 then the excess value should be added with KB. When sum of the KBs exceeds 1024 then excess value should be added with MB, similarly when sum of MBs exceeds 1024 then excess value should be added with GB. In case GB value exceeds 1024 then simply discard the excess value.

Overload >= operator to verify whether object1 >= object2.
object1 is greater than object2 only if GB of object1 > GB object2 and MB of object1 > MB of object2 and KB of object1 > KB of object2 and bytes of object1 > bytes of object2.
object1 is equal to object2 if GB of object1== GB object2 and MB of object1== MB of object2 and KB of object1 ==KB of object2 and bytes of object1 == bytes of object2

Use parameterized and non-parameterized constructors to initialize the objects.

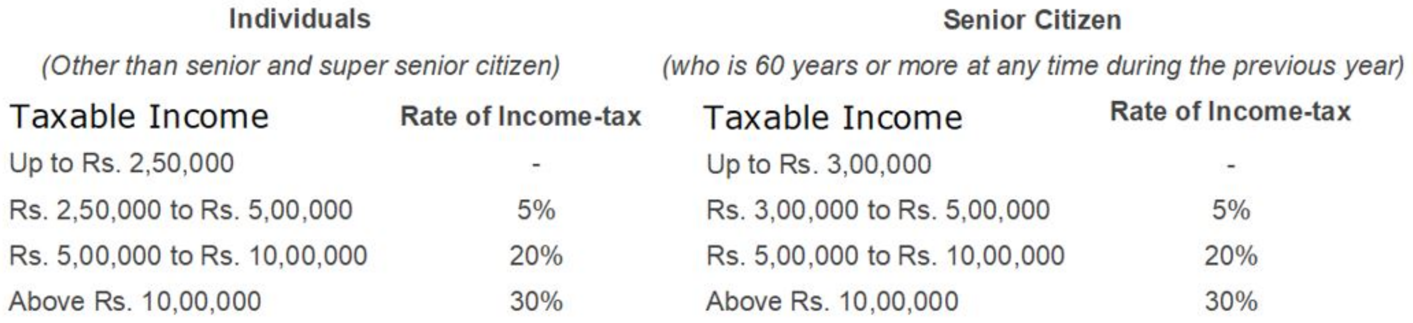
Question 2 (25 Marks)

Implement an abstract base class called ***IncomeTaxPayee*** consisting of the data members: Name, Gender, Age, Gross Pay, Deductions, Taxable Income and Tax on Taxable Income.

Derive three classes as ***Individuals***, and ***SeniorCitizen*** from ***IncomeTaxPayee***. In each class implement a function called ComputeTax with the appropriate tax formula. This function should be pure virtual in the base. Use the base class pointeer to invoke the ComputeTax function of the derived classes. Display the tax details for derived class objects.

Note: Taxable Income = Gross Pay – Deductions.

Tax on Taxable Income to be computed using the following conditions.



Individuals

(Other than senior and super senior citizen)

Taxable Income	Rate of Income-tax
Up to Rs. 2,50,000	-
Rs. 2,50,000 to Rs. 5,00,000	5%
Rs. 5,00,000 to Rs. 10,00,000	20%
Above Rs. 10,00,000	30%

Senior Citizen

(who is 60 years or more at any time during the previous year)

Taxable Income	Rate of Income-tax
Up to Rs. 3,00,000	-
Rs. 3,00,000 to Rs. 5,00,000	5%
Rs. 5,00,000 to Rs. 10,00,000	20%
Above Rs. 10,00,000	30%