**Experiment: 10**

PART B

(PART B: TO BE COMPLETED AND SUBMITTED BY STUDENTS)

Students must execute all the programs, write executed code in the workbook, and submit part B of experiment 8 on the student portal. The filename should be **PPS\_batch\_rollno\_experimentno. Example: PPS\_A1\_A001\_P8**

|  |  |
| --- | --- |
| **Roll No.:** | **Name:** |
| **Prog/Yr/Sem:** | **Batch:** |
| **Date of Experiment:** | **Date of Submission:** |

**Aim:** Programming using object-oriented programming (using data members, member functions, constructors, overloading & inheritance)

**Tasks:**

|  |  |
| --- | --- |
| Sr. No. | Problem Statement |
| 1 | Class “Employee” has data members: Emp\_id, Emp\_name and Emp\_sal and this class uses a parameterized constructor to accept the details of 2 employees and display the results using the display () function. |
| 2 | Define class Complex with real and imaginary as data members, define default and parameterized constructor to initialize two complex numbers. Define add (Complex, Complex) member function to add two complex numbers and show( ) function to display both the complex numbers with their addition. |
| 3 | Rewrite above question to add two complex numbers using overloaded + operator. |
| 4 | Create a class rectangle with (length, width), derive a class box with additional member (depth). In both the classes write parameterized constructor to initialize data member and area ( ) function to find area. Define main ( ) and create appropriate objects to call area ( ) function. |
| 5 | Declare a class employee having data members as id, name and member function as getData() and Display(). Derive class manager from employee class. Manager class has data members as salary and member functions as getdata() and display(). Again derive class project manager from manager. Project manager class have data members as total, experience, number of projects handled and member functions as getdata() and display(). Write a program using multilevel inheritance to display all details of project manager. |

**Executed Code, Input and Output**

|  |  |
| --- | --- |
|  | Class “Employee” has data members: Emp\_id, Emp\_name and Emp\_sal and this class uses a parameterized constructor to accept the details of 2 employees and display the results using the display () function. |
| **Executed Code: -**  // Paste the executed code here  **Input Output: -**  // Paste the input/output of executed code | |
|  | Define class Complex with real and imaginary as data members, define default and parameterized constructor to initialize two complex numbers. Define add (Complex, Complex) member function to add two complex numbers and show( ) function to display both the complex numbers with their addition. |
| **Executed Code: -**  // Paste the executed code here  **Input Output: -**  // Paste the input/output of executed code | |
|  | Rewrite above question to add two complex numbers using overloaded + operator. |
| **Executed Code: -**  // Paste the executed code here  **Input Output: -**  // Paste the input/output of executed code | |
|  | Create a class rectangle with (length, width), derive a class box with additional member (depth). In both the classes write parameterized constructor to initialize data member and area ( ) function to find area. Define main ( ) and create appropriate objects to call area ( ) function. |
| **Executed Code: -**  // Paste the executed code here  **Input Output: -**  // Paste the input/output of executed code | |
|  | Declare a class employee having data members as id, name and member function as getData() and Display(). Derive class manager from employee class. Manager class has data members as salary and member functions as getdata() and display(). Again derive class project manager from manager. Project manager class have data members as total, experience, number of projects handled and member functions as getdata() and display(). Write a program using multilevel inheritance to display all details of project manager. |
| **Executed Code: -**  // Paste the executed code here  **Input Output: -**  // Paste the input/output of executed code | |

**Observation and Learning: -**

* Write your observation and learning