University School of Information, Communication & Technology

List of Exercises

Paper: Object Oriented Programming Using C++ Lab. (IT-255)

1. Design a class to represent a bank account. Include the following members:
   Data members:
   -Name of depositor
   - Account Number
   -Type of Account
   -Balance Amount in the account
   Methods:
   - To assign initial values
   - To deposit an amount
   - To withdraw an amount after checking balance
   - To display the name & balance
2. Write a program in C++ to demonstrate the difference between a 'Call by Value', 'Call by Reference' and 'Call by Address'.
3. Write a program in C++ to calculate "n" to the power "p" using default value parameter for "p".
4. Write a program in C++ to create and sort an array. The array should be created dynamically.
5. Write a class **CAccount** which contains two private data members to store account number and balance amount and three member functions:
   a. A constructor that allows the user to set initial values for both the data members and a default constructer that prompts for the input of the values for the above data members.
   b. A function called inputTransaction, which reads a character value for transactionType ('D' for deposit and 'W' for withdrawal) and a floating-point value for transactionAmount, which updates balance amount. The function should check for the balance amount for withdrawal.
   c. A function called printBalance which prints on the screen the account number and balance amount.
6. Design and implement a class **Distance** with two private data members to store the distance in feets and inches. Include the methods in class to perform the following operations:
   a. Blank Constructor
   b. Parameterized Constructor
   c. To take input from the user
   d. To adjust the value of inch in a way that the value of inch must be less than 12.
   e. To display the distance in the form 5' 6".
7. Write a program that reads a sequence of names, one per line, and then sorts and prints them.
8. Design and implement a class **String** using dynamic memory allocation operators with one private data member, a character pointer pointing to a variable length string. Include the methods in class to perform the following operations:
   a. Default Constructor and Copy Constructor

b. Destructor
c. To take input from the user
d. To display the string
e. To return the reverse of stored string

9. Create two classes **DistanceFI** and **DistanceMC** which store the value of distances. DistanceFI stores distances in feets and inches and DistanceMC stores distances in meters and centimeters.
   a. Write methods in both classes to read values for the class objects.
   b. The display should be in the format of feet and inches or meters or centimeters depending on the object on display. Overload the '+' operator to add the two objects of the same class.
   c. Overload the '+' operator to add the one object of DistanceMC with another object of DistanceFI.
   d. Overload the assignment operator to stores the result either in a DistanceFI object or DistanceMC object depending receiving object.
   e. Write the appropriate methods to allow Basic to Class, Class to Basic and one Class to another Class type conversion.

10. Design and implement a class **Complex** to store complex numbers. Include the methods to allow the following operations:
   a. Constructors to allow initialization of objects in different manner.
   b. To take input
   c. To display the complex number in the form a + ib;
   d. Overload the operators '+, -, /, *' to allow the arithmetic operations on complex numbers.
   e. Overload all relational operators using friend functions.