Slot :B2

DA2

BCSE102L Structured and Object-Oriented Programming

Faculty In charge : Dr.J.Uma Maheswari

1. Suppose that we want a class that can be used to represent people at a university (e.g., faculty, staff, undergraduate students, and graduate students). For all types of people, we want to record their name and university ID. For specific types of people, we want to record some additional information. In the case of faculty, we want to record their rank (i.e., Assistant Professor, Associate Professor, or Full Professor). In the case of undergraduate students, we want to record their year of study (i.e., 1, 2, 3, or 4). In the case of graduate students, we want to record their supervisor's university ID. Suggest a class hierarchy that might be used to represent the above collection of people and implement the same.

2. Create a class Book and define display method to display book information. Inherit Reference_Book and Magazine classes from Book class and override display method of Book class in Reference_Book and Magazine classes. Make necessary assumptions required.

3. Sunrise Basket founder has decided to organize a fun event at your college. The event coordinator has announced a coding contest for creating the application for the Contest. The Best application would be used for the fair and the developer gets a cash prize. You are a well versed and aspiring Programmer in your college. Many programmers have enrolled themselves for the contest and you are one of them. Every contestant is provided with a Schema diagram of the Fair. Get yourself acquainted with Schema and brace yourself for the challenge!!!. As a part of this, the Application requires a user prompt to create a new Item type. Hence create an ItemType class with the following private attributes. • name (String) • deposit(Integer) • costPerDay(Integer) Include appropriate Getters and Setters for the class and also include a method "void display()" to display the output shown in the sample output. The main is implemented to get input from the user and display. Write the suitable code complete ItemType class.

4. Develop a program for banking system for account management. Each account has the following attributes: AccountID, HolderName, and Balance. Declare one constructor with three parameters that initialize the three attributes to some default values. Attributes must be validated.
AccountBalance must be greater than or equal to zero. If not, it is set to zero.
AccountID must be between 100 and 999. If not, set it to -1 to indicate that it is invalid.
Use the name setAccountBalance (…) to modify the account balance. Write one method credit() to deposit money into the account. The method should return the new balance after the money deposit.

5. Given inputs are N (vary for each year), dummy number and a number k (vary for each course), check if the number is a valid dummy number or not. The university has kept the dummy numbers for the booklets in such a way that the sum of the dummy number digits has to be divisible by either 2 or 3 or 5. And also divisible by k. All the valid dummy numbers are of N digits and it should be a positive number in addition to the above conditions. Write a C++ program to do the task using classes, objects, constructors and destructors.

6. Explain the significance of static data members. Design an account class with members: account number, amount and interest rate. Interest rate has to be declared as static data member. Include member function to calculate monthly interest rate as (amount*interest rate)/12, add the monthly interest to amount whenever the function is invoked.

7. For an Insurance application, create a C++ class named AutoInsurance that can hold a driver's age, name, state of residence and a unique Insurance ID. The premium base price is Rs.1000 for residents of Kerala (KE) and Rs.500 for residents of TamilNadu (TN). Additionally, each driver pays 2 times the value of 1000 minus his or her age. If the driver is younger than 16 or older than 80, or not a resident of TamilNadu or Kerala, display a message "Invalid input". Create 3 driver objects for the class and calculate their premium. Use the default constructor to initialize the age of the driver to 0. Use static data member to count the number of objects. Use a destructor to print the message that is given in the sample output.

   Sample input

   25 // age of Driver1
   Hari //name of driver1
   TN // state of residence of driver1
   1234 //insurance id
   40
   Mary
   KE
   2345
   15
   Anu
   KE
   3456

   Sample output

   Driver 1:2450
   Driver 2:2920
   Driver 3: Invalid input
   No of objects:3
   Anu is deleted
   Mary is deleted
   Hari is deleted