JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY–NOIDA
CI311- Object-Oriented Programming
Tutorial / LAB – Week 9[27Oct to 1 Nov]

Practice Set
Inheritance and Polymorphism

Q1.(A) Imagine a publishing company that markets both book and audiocassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int), and tape, which adds a playing time in minutes (type float). Each of these three classes should have a proper constructors, destructors and display functions to display its data. Write a main() program to test (a) order of constructor and destructor invocation, (b) the book and tape classes by creating instances of them, asking the user to fill in data and then displaying the data.

(B). Start with the publication, book, and tape classes of 'Q1'. Add base class sales that hold an array of three floats so that it can record the dollar sales of a particular publication for the last three months. Include a function to get three sales amounts from the user, and a function to display the sales figures. Alter the book and tape classes so they are derived from both publication and sales. An object of class book or tape should input and output sales data along with its other data. Write a main() function to create a book object and a tape object and exercise their input/output capabilities.

(C). Assume that the publisher in 'A and B question' decides to add a third way to distribute Books: on computer disk, for those who like to do their reading on their laptop. Add a disk class that, like book and tape, is derived from publication. The disk class should incorporate the same member functions as the other classes. The data item unique to this class is the disk type: either CD or DVD. You can use an enum type to store this item. The user could select the appropriate type by typing c or d.

(D). Start with the publication, book, and tape classes of 'A question'. Suppose you want to add the date of publication for both books and tapes. From the publication class, derive a new class called publication2 that includes this member data. Then change book and tape so they are derived from publication2 instead of publication. Make all the necessary changes in member functions so the user can input and output dates along with the other data.

Q2.Assume that a bank maintains two kinds of accounts for customers, one called as savings account and other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders also maintain a minimum balance and if the balance falls below this level, a service charge is imposed.

Create a class account that stores customer name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specificnto their requirements. Include necessary member functions in order to achieve the following tasks:

a)-Accept deposit from a customer and update the balance

b)-Display the balance.
c)-Compute and deposit interest.
d)-Permit withdrawal and update the balance.
e)-Check for minimum balance, impose penalty, necessary and update the balance.

Q3. (A)

```
#include <iostream>
using namespace std;
const int LEN = 80; //maximum length of names


class student //educational background
{
private:
char school[LEN]; //name of school or university
char degree[LEN]; //highest degree earned
public:
void getedu()
{
cout << " Enter name of school or university: ";
cin >> school;
cout << " Enter highest degree earned \n";
cout << " (Highschool, Bachelor's, Master's, PhD): ";
cin >> degree;
}
void putedu() const
{
cout << "\n School or university: " << school;
cout << "\n Highest degree earned: " << degree;
}
};

class employee
{
private:
char name[LEN]; //employee name
unsigned long number; //employee number
}
```

Derive a class called employee2 from the employee class in the above EMPLOY program. This new class should add a type double data item called compensation, and also an enum type called period to indicate whether the employee is paid hourly, weekly, or monthly. For simplicity you can change the manager, scientist, and laborer classes so they are derived from employee2 instead of employee. However, note that in many circumstances it might be more in the spirit of OOP to create a separate base class called compensation and three new classes manager2, scientist2, and laborer2, and use multiple inheritance to derive these three classes from the original manager, scientist, and laborer classes and from compensation. This way none of the original classes needs to be modified.

3(B). There is only one kind of manager in the above program. Any serious company has executives as well as managers. From the manager class derive a class called executive. (We'll assume an executive is a high-end kind of manager.) The additional data in the executive class will be the size of the employee's yearly bonus and the number of shares of company stock held in his or her stock-option plan. Add the appropriate member functions so these data items can be input and displayed along with the other manager data.

Q4. Write a program that uses the class ***safe_int_array*** which this class will check if the index is valid when call its get/set function. Write another
class called ***idx_int_array,*** which is derived from ***safe_int_array.*** Class
idx_int_array provides methods to access an array which its index starts
at a designated number. For example, idx_int_array a(-10,2), a.get(-10)
will return the array's first element and a.get(2) will return the array's last
element. Because of class ***safe_int_array***, a.get(-11) will print error message and return a value invalid.
safe_int_array:
safe_int_array sa(10); // an int array with 10 elements
sa.set(i,100); // sa[i]=100, if i is invalid, error message will be printed;
sa.get(i); //return sa[i], if i is invalid, error message will be printed and
return -1;
idx_int_array:
idx_int_array ia(-10,2); // an int array which its index is between -10 and 2.
ia.set(-10,100); // set the first element to 100
ia.get(-10); //return 100

Q5. Create the abstract class **Animal** and a set of two or three animal sub-classes such as **Tiger** and **Lion. Details for animal class is-**

- **char* getName()** - which returns the name of the animal.
- **int getAge()** - which returns the age in milliseconds.
- **void talk()** - which causes the animal to display on the screen their name, age and what type of animal they are.
  Create a Zoo class containing an array of references to animals with the functionality below-

The Zoo provides the following methods :-

- **boolean addAnimal(Animal newAnimal)** - which adds a new animal to the Zoo.
- **void displayAnimals()** - which displays a list of the type and name of all animals in the zoo by cage number.

- **void feedingTime()** - which causes all animals in the zoo to talk one after the other.
- **Animal& selectCage(int cageNum)** - which returns a reference to the animal in a specified cage.

  The main program has following functionalities-

  Add a new animal to the zoo - first prompting for name and type and creating the animal.

  Display all animals currently in the zoo.

  Display the name and age of the animal in a selected cage. Trigger feeding time.

  Implement a menu-driven main program which uses the Zoo and the various animal classes and test it.

Q6. **D**esign a base class name employee. The class should keep the following information in member variables:

Employeename

Social security number, in the formatXXX-X-XXXX which each X is a digit within the range 0-9.

Employee number, in the format XXX-L,where each X is a digit within the range 0-9, l is a letter within the range a-m.

Hiredate;-

And a constructor ,destructor and other appropriate member function to the class. the constructor should dynamically allocate enough memory to hold the employee name and the destructor should free the unused memory.

Next,design a class name employeepay . this class should be drived from the employee class. It should keep the following information in member variables:-

Annual pay

Monthly pay

Dependent(the no. of dependents employee claims). Demonstrate the class in a program that ask the user to enter sample data , and than display it on the screen.

Q7 (A) Design a class called miltime that is derived from the time class. The miltime class should convert time in military(24-hour) format to the standard tie format used by the time class. The class should have the following member variables:-

milhours:-contains the hour in 24-hour format. For example, 1:00 p.m would be stored as 1300 hours , and 4:30 p.m would be stored as 1630 hours.

<u>Milseconds:-</u>contains the second in standard format

The class should have the following member functions:-

<u>Miltimes:-</u> the constructor should accept arguments for the hour and seconds, in military format the time should than be converted to standard time and store in the hours, minute and second variable of the time class.

<u>Settime:-</u>except arguments to be stored in the milhours and milseconds variables. The time shoul than be converted to standard time and store in the hours, minute and second variable of the time class.

<u>Sethours:</u>-returns the hour in military format.

<u>Getstandhours:</u>-returns the hour in standard format.

Demonstrate the class in a program that ask the user to enter the time in military format.the program should than display the time in both military and standard format.

7 (B) Design a class name timeclock. The class should be derived from the miltime class you design in ques 3.the class should allow the programmer to pass two times to it ”starting time and ending time”. the class should have a member function that returns the amount of time elapsed between the two times. For example, if the starting time is 199 hours(9:00 a.m) and the ending time is 1300 hours(1:00 p.m), the elapsed time is 4 hours.

<u>Q8.</u>Design a class called studentinfo. It should have member variables for the following information:

Student name

Student id(12 characters)

Major(computer science, business etc.)

Write appropriate member function to store and retrieve information in the member variables . the constructor should dynamically allocate enough memory to hold the student name. the destructor should free the memory.

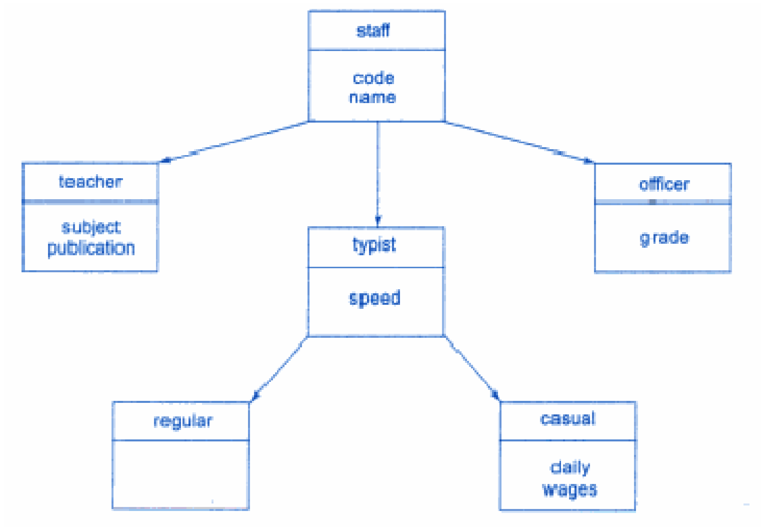
Design another class called grades. This class should be derived from the studentinfo class. It should have the member variables for the following information:-

Test grades(the class should hold 10 test grades)

Test average

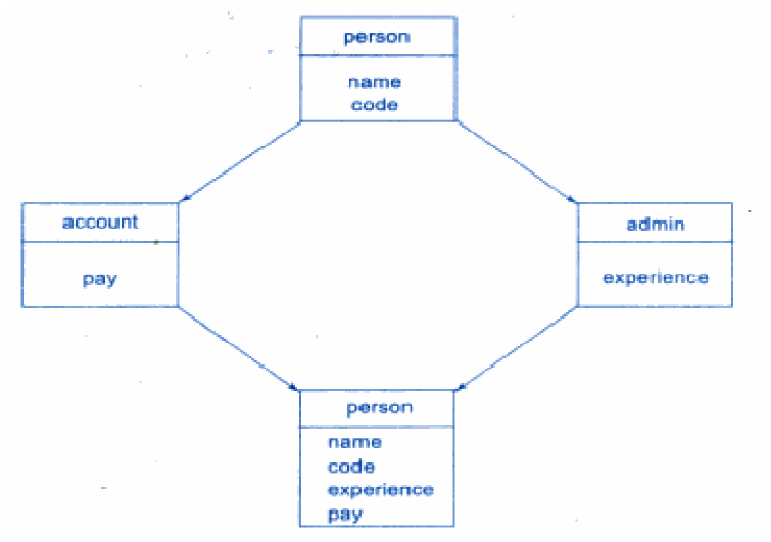
Write appropriate member function to store and retrieve information in the member variables. Demonstrate the classes in a program that declares and array of grades object. The user should enter the grades for each student, and the program should display each student average.

Q9.(A) An educational institute wishes to manage a database of its employees. The database is divided into a number of classes whose hierarchical relationships are shown in Fig. The figure also shows the minimum information required for each class. Specify all the classes and define the functions to create the database and retrieve individual information as and when required.



9 (B) The database created in problem 9(A) does not include educational information of the staff. It has been decided to add this information to teachers and officers (and not for typists) which will help management in decision making with regard to training, promotion, etc. Add another data class called education that holds two picies of educational information, namely, highest qualification in general education and highest professional qualification. This class should be inherited by the classes teacher and officer. Modify the program to incorporate these additions.

Q 10. Consider a class network of Fig. given below. The class master derives information from both account and admin classes which in turn derive information from the class person. Define all the four classes and write a program to create, update and display the information contained in master objects.



In problem 10 (B), the classes teacher, officer and typistare derived from the class staff. As we know, we can use container classes inplace of inheritance in some situations. Redesign the program of problem 4.3 such that the classes teacher, officer and typist contains the object of staff.

1. An **address book** consists of one or more contacts. The address book can have a maximum of 20 contacts. It also keeps track of the no of contacts.

   User can

   1. add contacts

   void add(contact c);

   2. search for a contact in the address book.

   Contact search(String keyword);

   To search a contact user gives a **keyword as a search parameter.**

   3. Displays all contacts

   Void display();

   Contacts can be either personal or and organizational. All types of contacts have **contact info** which consists of phone no, mobile phone and address. Personal contacts store the first name and the last name in addition to the contact info of that person. Organizational contact stores the name of the organization and its type (type of product/service which it deals with) in addition to the contact info. All types of contacts have a display function and **match function** which returns true if the keyword satisfies the matching criteria or else returns false. In case of personal contact the keyword satisfies the **matching criteria** if it matches **either with the first**

name or the last name. In case of organizational contact the keyword satisfies the matching criteria if **either it is a substring of the organization name or is matched with the type of the organization.**

Design and implement the most reusable object oriented solution for the above case. Your code should be designed in such a way that it allows duplication of code/reuses code written once and addition of new types of contacts with least amount of change required in the address book.

2. A petrol pump consists of 8 filling machines out of which 3 are single outlet machines and 5 are two outlet machines. Every filling machine has a **machine number.** (The naming convention for machine nos. is 1.0, 2.0, 3.0….). If the filling machine is a multiple outlet it has **two outlet numbers also** (The naming convention for outlet nos. is 1.1, 1.2 or 2.1, 2.2 etc….). The number before the decimal is same as it's the machine no and the digit after decimal specifies the outlet no.

   Ever filling machine has a **checkfree() function** which checks whether the machine is busy or free. This function generates a random number and based on the number generated it returns a double value.

   If the filling machine is **single outlet** checkfree function returns its machine no if free or else returns 0 if busy.

   Free if the number generated is between (0.5 and 1)

   Busy the number generated is between (0.0 and 0.5)

   If the filling machine **is two outlet** then it returns outletno of the outlet which is free or else 0 if both the outlets are busy, or the outlet no of the first outlet if bth free.

   Outlet 1 is free if number generated is between (0-0.25)

   Outlet 2 is free if number generated is between (0.25-0.5)

   Both the outlets are free if number generated between (0.5-0.75)

   Both busy if number generated between (.75-1)

A petrol manager class consists of all the filling machines. It has a function **process_ request** which returns the machine number/ outlet no of the machine which is free for filling. It has a function **addmachine** which allows you to add

A new machine to the manager.

Note: Use the following class to **generate random numbers**

Class name: Math

Function declaration: Public static double random();

This function returns a value between 0 and 1.

Design and implement the most reusable object oriented solution for the above case. Your code should be designed in such a way that it allows duplication of code/reuses code written once and addition of new types of fillingmachines with least amount of change required in the petrolpump manager.

3. You need to store information for bank accounts. For purposes of the problem, assume that you only need to store the current balance, and the total number of transactions for each account and an account no.

An account needs to respond to the following

messages:

Constructor

Initialize a new account

void Withdraw(float amt)

subtract the amt to the balance and increment the number of transactions

float Balance();

return the current balance

void Display()

print out the account monthly summary

There are **two types of accounts:**

TypeA: deposit and withdraw just affect the balance. At the end of each month There is a $5.00 monthly fee which counts as a normal withdrawal transaction fee is deducted from the balance.

Type2: Each withdrawal generates a $0.50 fee.

You have an **accounts manager class** which keeps track of all the accounts in the bank. It responds to the following messages:

Void addaccount(account a);

Adds a new account

Void Withdrawlfromaccount(double amt, int accno)

Searches for the account with the given accountno and withdraws amt from that account;

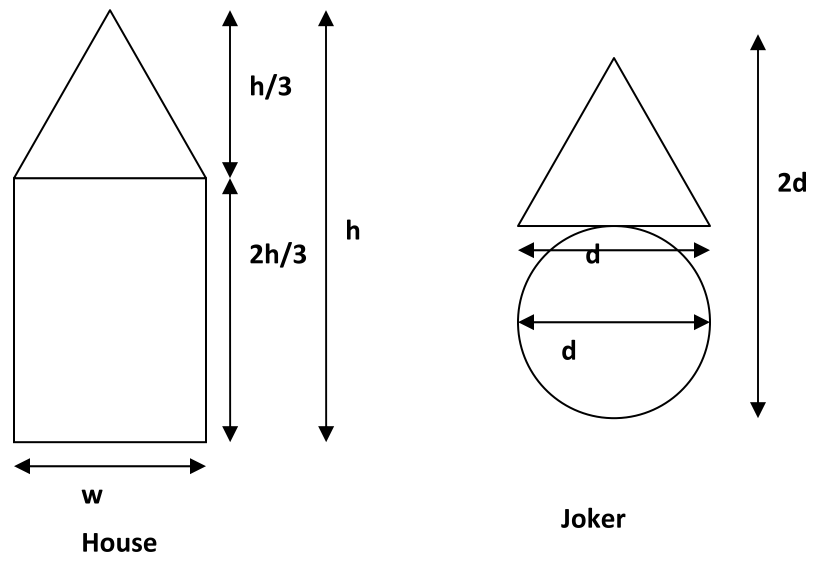
Void printallbalance()

Prints the accountno, balance, no of transactions of all accounts present in the bank manager class.

Design and implement the most reusable object oriented solution for the above case. Your code should be designed in such a way that it avoids duplication of code written once and allows addition of new types of accounts with least amount of change required in the bankmanager class.

4. ABC company is a company that takes orders of painting the homes of people. The company paints shapes on the walls of their homes. The shapes can be a house or a joker. The rate of

painting a house is Rs100 per sq centimeter and for joker it is Rs150 per sq centimeter . The cost of painting the shape is calculated by multiplying the rate by the shape with its surface area. For all shapes the customer needs to specify the dimensions of the shape. For the joker the customer specifies the diameter d as shown in the figure. For the house it needs to specify the width(w) and the height(h) as shown. The company wants to make an order processing system which stores the orders for painting and displays the cost of each order. The user can add any no of shapes to an order. For each shape the system stores the dimensions of the shape. There is a calcost() function for order which calculates the cost of painting the complete order.



Design and implement the most reusable object oriented solution for the above case. Your code should be designed in such a way that it avoids duplication of code written once and allows addition of new types of shapes with least amount of change required in the order class.