

Tutorial Sheet – Odd Sem 2018

Data Structure Tutorial

Week - 1 (17-Jul to 22-Jul)

Topics: Object oriented concepts and programming using C++

Q1 You are required to create a class — “Invoice”. This class might be used by a departmental store to represent an invoice for an item sold at the store. An Invoice should include four data members— item number (type string), item description or name (type string), quantity of the item being purchased (type int) and price per item (type int). Your class should have a constructor that initializes the four data members. Provide a set and a get function for each data member. In addition, provide a member function named `getInvoiceAmount` that calculates the invoice amount (i.e., multiplies the quantity by the price per item), then returns the amount as an int value. The class should be able to give the count of all the objects which are created through copy operation. Write a function that accepts two invoices for different departmental stores and return the maximum quantity out of two invoices. Write a test program that demonstrates class Invoice’s capabilities.

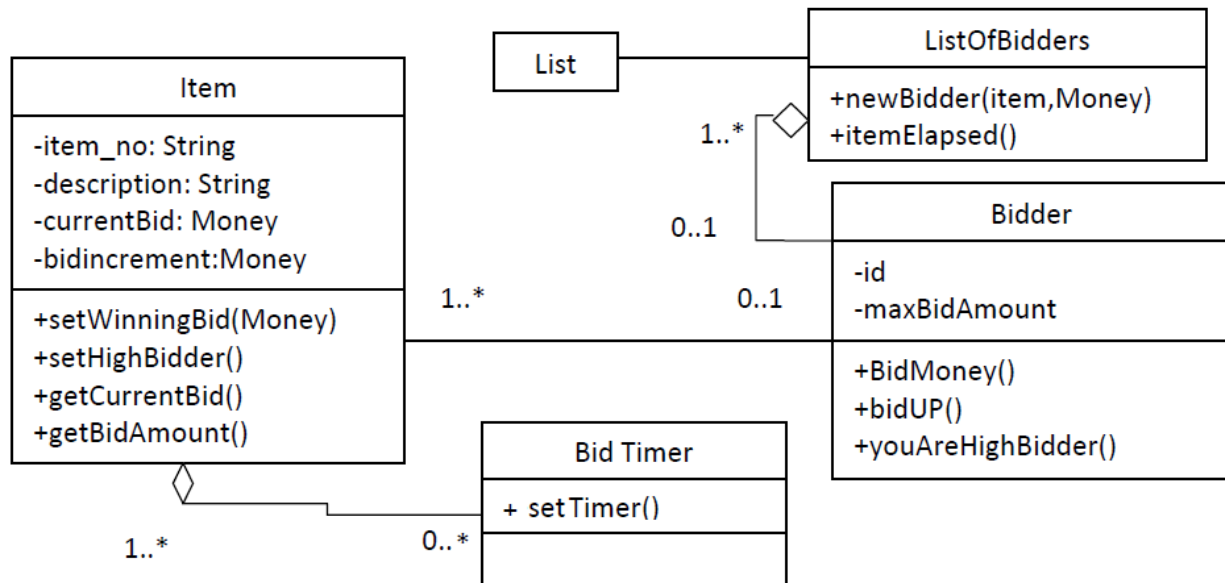
Q2 Create a class —Employee that includes three pieces of information as data members—a first name (type string), a last name (type string) and a monthly salary (type int). Your class should have a constructor that initializes the three data members. Provide a set and a get function for each data member. Create two Employee objects and display each object’s yearly salary. Then give each Employee a 10 percent raise and display each Employee’s yearly salary again. The class should be able to give the count of all the default objects. Write a function which takes two employees and return the name of the employee with higher salary. Write a test program that demonstrates class Employee capabilities.

Week - 2 (23-Jul to 28-Jul)

Topics: Object oriented concepts and programming using C++

Q1. A class diagram is shown in figure below. Implement the classes, appropriate relationships, and respective member functions shown in the class diagram. If needed, define member functions in appropriate classes to answer following queries:

- (a) Find the item and Bidder with highest bid out of a stock of 10 items
- (b) List the items with current bid amount (if any).



Q2. Mobile phones are manufactured by many companies, e.g. Samsung, Sony, etc. Usually performance of these mobile phones are measured by memory (in GB), front camera (in MP), rear camera (in MP), and battery backup (in Hr) and accordingly price of the mobile phones varies. To operate mobile phones, users need sim-card which is to be issued by service provider, e.g. Airtel, Vodafone, etc. Besides the storage capacity, each sim is uniquely identified by the Mobile Number. When a sim is placed into a mobile, then only a mobile is functional otherwise not. Users need to purchase mobile as well as sim to make calls, use internet, etc. In each such phone, user can store contact details of his/her friends in the mobile storage or sim storage. To make a call, user is required to search the mobile number of his/her friend from the contact list (either of phone or of sim or of both). Whenever a call is made from one user (say X) to another user (say Y), it is required to display the message —Airtel/Vodafone/etc. user with mobile no. XXXXXX is calling on the mobile screen of Y. Record of this call is to be stored into both users mobile phone: for X, it is outgoing and for Y, it is incoming. Further, a user may have maximum 5 bank accounts, where he/she can perform the transactions (either deposit or withdrawal). Withdrawal is based on the available balance in an account of the user. Based on the available balance in all accounts of a user, he/she decides whether a specific mobile can be purchased or not. Draw the class diagram and implement it in C++ and answer following:

(a) Who are the users capable enough to purchase a specific mobile phone model?

(b) Name the user who has made maximum call to a user (say Y, here Y can be called by many user).

Q4. A grocery shop stores a variety of products which are identified through product_id, prize, name, manufacturing date. A product can be consumable & non consumable. A consumable product in addition is also having the expiry date. A grocery shop is having a no. of customers

identified by the name & address. The customers can be members & non members. The members are having unique membership id while the non members are having a unique 4 digit mobile no. Grocery shop gives a discount of 20% on the total bill to the members & 2% & 4% on consumable & non consumable goods to the non members. Use the concept of inheritance & overloading to implement the above scenario. Total discount availed at the end of a day is required to be generated assuming 10 customers shopping for at max 5 products. Display the customer detail that has availed the maximum discount.

Week - 3 (30-Jul to 04-Aug)

Topics: Object oriented concepts and programming using C++

Q1. Four classes, C1, C2, C3, and C4 have been given in following figures along with main(). Get the output of the C++ program without running the code.

<pre>class C1 { int x; public: C1(){ x = 50; cout<<"\n Print 1"; } C1(int temp){ x = temp; cout<<"\n Print 2"; } void f1(){ cout<<"\nEnter X"; cin>>x; } void f2(int temp){ x = temp; } void f3(){ cout<<"\nX in C1 is "<<x; } ~C1(){ cout<<"\n Print 3"; } }; class C2 { int y; public: C2(){ y = 60; cout<<"\n Print 4"; } C2(int temp){ y = temp; cout<<"\n Print 5"; } void f4(){ cout<<"\nEnter Y"; cin>>y; } void f5(int temp){ y = temp; } void f6(){ cout<<"\nY in C2 is "<<y; } ~C2(){ cout<<"\n Print 6"; } };</pre>	<pre>class C3 { C1 *objC1type[2]; C2 objC2type[2]; public: C3(){ cout<<"\n Print 7"; } C3(int temp){ cout<<"\n Print 8"; } void f7(int temp){ objC1type[0] = new C1; objC1type[1] = new C1(temp); } void f8(){ C1 *temp1 = new C1(20); C2 temp2; temp1 -> f3(); temp2.f6(); } void f9(C1 *temp1) { C1 *temp2, *temp3, temp4; temp2 = temp1; temp2->f2(30); temp3 = new C1; temp3->f2(40); temp4.f2(70); temp1 -> f3(); temp2 -> f3(); temp3 -> f3(); temp4.f3(); } ~C3(){ cout<<"\n Print 9"; } }; class C4 { C3 *obj1, obj2; public: C4(){ obj1 = new C3; cout<<"\n Print 10"; } void f10(){ C3 *temp1, temp2; temp1 = new C3(10); } ~C4(){ cout<<"\n Print 11"; } };</pre>
<pre>int main() { C1 *objC1 = new C1; objC1 -> f3(); C3 objC3; C4 objC4; objC3.f7(5); objC3.f8(); objC3.f9(objC1); objC4.f10(); objC1 -> f3(); return 0; }</pre>	

Q2. A C++ program along with main() is presented in following figure. Analyze the output of the program without running the code.

```

class a
{
    protected: int x;

    public:
        a() { x = 10; cout<<"n1"; }  a(int t) { x = t; cout<<"n2"; }  ~a() { cout<<"nA is deleted"; }
};
class b : public a
{
    protected: int y;

    public:
        b() { y = 20; cout<<"n3"; }  b(int t1, int t2):a(t1) { y = t2; cout<<"n4"; }  ~b() { cout<<"nB is deleted"; }
};
class c : public a
{
    protected: int z;

    public:
        c() { z = 30; cout<<"n5"; }  c(int t1, int t2):a(t1) { z = t2; cout<<"n6"; }  ~c() { cout<<"nC is deleted"; }
};
class d : public b, public c
{
    int m;
    public:
        d() { m = 40; cout<<"n7"; }
        d(int t1, int t2, int t3, int t4, int t5): b(t1,t2), c(t3,t4) { m = t5; cout<<"n8"; }
        void f1() { cout<<"n\n"<<b::x<<"t"<<c::x<<"t"<<y<<"t"<<z<<"t"<<m; }
        ~d() { cout<<"nD is deleted"; }
};
int main()
{ d o2, o3(1, 2, 3, 4, 5); d *o1=new d(1, 2, 3, 4, 5); cout<<"nSize of O1="<<sizeof(o1); o1->f1(); delete o1; cout<<"nSize of O2="<<sizeof(o2);
return 0; }

```

Q3. A C++ program along with main() is presented in following figure. Analyze the output of the program without running the code.

```

class a
{
    public:
        int x;
        a() { x = 10; };
        a(int t) { x = t; }
};
class b : public a
{
    public:
        int y;
        b() { y = 30; x = 40; }
        b(int t) : a(t+10) { y = t; }
};

int main()
{
    a *o1 = new a, o2(20); b o3, *o4 = new b(50); o4 = (b *)o1; cout<<o4->x; return 0;
}

```

Q4. It is required to create a book information system, where record of books is maintained. Any book can be categorized by the title of book, author (s) of the book, price of the book, publication year of the book, and pages. A page of a book contains the page no, texts, Figure. No, and Table No. A page may or may not have any text/figures/tables (i.e. a page may have only texts, or only figures, or only tables, or figures and tables, or figures, tables, and texts, or

texts and tables, etc.). There can be maximum 3 figures in a page (if there are no texts or tables). Similarly there can be maximum 3 tables in a page (if there are no texts or figures), maximum 3 tables & figures (if there are no texts). Further, any book will have minimum 5 pages and maximum 100 pages. Every book will have an issue card where issue details (issue date, expected return date, student id, and return date) can be entered while issue or return. There can be maximum 30 entries possible in this card (i.e. total 30 records of issue/return can be stored in the book). Maximum 5 books can be issued to a student (categorized by student id, name, branch), if issued book count at any stage for a student is 5 or overdue (i.e. Exceeded the expected return date) then warning will be issued to the student and fine of the overdue books are to be imposed @2 Rs. per day. Create necessary classes for the above scenario and draw the class diagram. Write a test program that demonstrates above scenario and answer following queries:

- (a) Identify which book having maximum number of figures has been issued maximum time, and
- (b) Identify the name of student on which maximum fine is imposed on current day.