

## PROGRAMMING EXERCISES

---

### LAB 8: ACCOUNT

Account
accountName : char[] accountNumber : char[] accountType: int accountBalance : float
getBalance() : float withdraw(): void deposit() : void printDetail() : void

Write a complete C++ program that is able to do the following tasks:

1. Write a class definition based on the class diagram above. Declare the data members as private, while the member functions as public.
2. Define the constructor for class account by one of the following method:
  - i. Overload constructors - Define the first constructor without parameter and the second constructor with parameter. Both constructor will initialize the data members with suitable values.
  - ii. Constructor with default argumen - initialize the data members with suitable values.
3. Write implementation function for the following member function:
  - i. **getBalance( )** – return the the account balance.
  - i. **withdraw( )** – allow user to withdraw some money. Check whether the balance in the account is sufficient to be withdrawn.
  - ii. **deposit( )** – allow user to deposit some money and update the balance in the account.
  - iii. **printDetail( )** – print all the information in the account.
4. Write **main()** program that will declare 2 instances of object account. Declare the instances based on the parameters provided in the constructor.
5. Implement all member functions : **getBalance( )**, **withdraw( )**, **deposit( )** and **printDetail( )** which are accessed through the objects.

6. Divide the program into header files and implementation files. Compile cpp files, link and create project file.
7. Declare an array of account with size 10. Read information about the account from a data file.
8. Implement the member functions through the **Account** array elements.

### LAB 9: TEST GRADING PROBLEM

1. Write a program to grade a test and print the student's score. The questions are in multiple choices form. There are 25 questions given in the test and each question has 4 choices; A, B, C and D. Assumed that there are 20 students taking the test. Each answer is stored in an array named answer as given below:

```
char answer [ ] = {"BACDACBDABBACADAABACABCD"} ;
```

The student's name, matrix number and the student's answer is provided in a data file in the following format:

/* Matrix No	Student's answer */
11111	ABCDABCDABCDABCDABCDABCD
11122	BBCCDDAABBCCDDAABBCCDDAAB
11133	AAAAABBBBCCCCDDDDABCD
:	:
:	:

For every correct answer, 4 marks will be given as a score and for every wrong answer, One (1) mark will be deducted from the score.

Grade will be given base on the following scheme:

Marks/Markah	Grade/Gred
75 – 100	A
65 – 74	B
55 – 64	C
45 – 54	D
0 – 44	E

The output for the program is a list with the student's name, the matrix number, the score and the grade for each student.

**LAB 10: INVENTORY PROBLEM**

Inventory
<pre>itemName : char[] itemKod : char[] itemPrice: float qtyInStock : int totalEarned: float</pre>
<pre>getData() buy() addStock() discountPrice() printDetail()</pre>

Write a complete C++ program that is able to do the following tasks:

1. Write a class definition based on the class diagram above. Declare the data members as private, while the member functions as public.
2. Write implementation function for the following member function:
  - i. **getData ( )** – read value for data member.
  - ii. **buy(int)** – deduct number of item in stock based on the integer value and add the total money earned to variable **totalEarned**.
  - iii. **addStock()** – allow user to add item in the stock.
  - iv. **discountPrice()** – set a new price based on the amount of the discount given.
  - v. **printDetail()** – print all the information in the **Inventory** instance.
  - vi. Provide appropriate constructor and destructor for the **Inventory** class.
3. Write **main()** program that will declare an array of **Inventory** and read information about the **Inventory** from a data file.
4. Implement all member functions through the element of the array.
5. Divide the program into header files and implementation files. Compile cpp files, link and create project file.

**LAB 11: FLIGHT CASE STUDY**

1. Given the following class diagram and main() program as follows:

Flight
destination no_Passenger Capacity status ticketPrice totalPayment
Constructor book(int, char[]); cancel(int); checkDetail();

1	//Program 2.30
2	<b>main()</b>
3	{
4	<b>Flight Flight1</b> ("Kuching",150,120.00,50);
5	<b>Flight Flight2</b> ();
6	<b>Flight1.book</b> (20,"senior citizen");
7	<b>Flight1.cancel</b> (2);
8	<b>Flight1.checkDetail</b> ();
9	:
10	}

Answer question a, b, and c by dividing the program into header file, implementation file and client file.

- a. Based on the class diagram, write the declaration of class `Flight`.
- b. Write source codes for the member function of class `Flight` with the following properties:
  - i. **Constructor with default argument** that receives 4 parameters that will initialize `destination`, `Capacity` and `ticketPrice` (You are required to set your own default argument.)
  - ii. **`book(int, char[])`** : a member function that receives 2 parameter, number of passenger who book the ticket and the status of the passenger. If the passenger is a "Senior Citizen", he will be given 50% discount for the ticket price. Update the total payment that the company will receive and update the number of passenger already booked the flight. Ensure that the number of passenger does not exceed the flight capacity.
  - iii. **`cancel(int)`**: This member function will receive number of passengers who want to cancel the booking. Update the number of passenger by deducting with the number of passenger that cancels the booking. All cancellation will only be refund by 50% of the price ticket. (Example: if the price of the ticket is RM200.00, the passenger will get back only RM100.00.) Update total earned.

iv. `checkDetail()`\_: Print information for all passengers.

c. Change `main()` program above to the following:

1	<code>//Program 2.31</code>
2	<code>main()</code>
3	<code>{Flight * Flight1;</code>
4	<code>(i)// Declare memory for Flight1 dinamically</code>
5	
6	<code>(ii)// Implement all operation in Flight class</code>
7	<code>using the pointer</code>
8	
9	<code>(iii)// Delete the memory pointed by Flight1</code>
10	
11	<code>}</code>
12	
13	

## EXERCISES

---

### EXERCISE 1: ADT and Class

For each of the following statement, determine **True** or **False**.

1. Data abstraction focuses on the operations of data and the implementation of the operations.
2. Abstract Data Type is a collection of data and a set of operations on the data.
3. The following declaration of `masa` class and overloading operator will lead to ambiguity error.

<pre>class masa {     : public:     masa(int, int=24, int=60);     masa(int); };</pre>
--

4. Assume that variable `myCar` is a pointer to an object of the `Car` class, and that the `Car` class has a member function named `accelerate()`. The following is a valid call to the `accelerate()` member function?

`Car->accelerate();`

5. Constructors do not have return type.
6. A constructor can have several parameters
7. A constructor has the same name as the class name.
8. A class can only have a single constructor.
9. A class can have several destructors.
10. The compiler will generate a destructor if the programmer does not provide one.
11. A programmer must provide a destructor for every class.
12. Destructor will be called once to destroy all instances of a class.
13. In C++, the declaration of a class should be placed in the Header file.

## EXERCISE 2: CLASS

Given the following class definition and functions prototype, answer the following questions.

```
1 //Program 2.11a
2 class Car
3 {
4     private:
5         string brand;
6         double price;
7         int CC;
8     public:
9         Car(){
10             brand="none";
11             price = 0;
12             C=0;
13         }
14
15         Car(string cb, double cp, int cc){
16             brand = cb;
17             price = cp;
18             CC = cc;
19         }
20
21         string getBrand(){
22             return brand;
23         }
24         double getPrice(){
25             return price;
```

```

25     }
26
27     void print(){
28         cout << "The brand of the car is : " << brand << endl;
29         cout << "The price of the car is : " << price << endl;
30         cout << "The CC of the car is : " << CC << endl;
31         cout << endl;
32     }
33 };
34 void fillData(Car cars[]); // non-member function
    Car cheapest (Car cars[]); // non-member function

```

- a. The following function is an application function of the above class. Describe what the function does.

```

1 //Program 2.11b
2 void fillData(Car cars[])
3 {
4     string cb;
5     double cp;
6     int cc;
7
8     int i=0;
9
10    for(int i=0; i<6; i++)
11    {
12        cout << "Please enter car brand for car no " << (i+1) << ": ";
13        getline(cin, cb);
14
15        cout << "Please enter car price for car no " << (i+1) << ": ";
16        cin >> cp;
17        cin.ignore();
18
19        cout << "Please enter car CC for car no " << (i+1) << ": ";
20        cin >> cc;
21        cin.ignore();
22
23        Car car(cb, cp, cc);
24        cars[i] = car;
25        cout << endl;
26    }
27 }

```

- b. Write an external function with the following function header. The function will find the cheapest car among the six cars read in the above **fillData()** function. Information of the six cars is passed as an array of object argument to the function. The function should return the cheapest car's object.

**Car cheapest (Car cars[])**

- c. Write program statements to print car information using the following array of object declaration and pointer declaration. The program statements request the user to enter an index between 1 to 6, and the program statements will print the

selected car's information using the pointer variable. For example, with the following information in the cars array, and if the user input index 3, the program will print the Proton Saga's information.

```
Car cars[6];  
Car *p;
```

	brand	price	CC
[0]	Honda	123000.00	2500
[1]	Mitsubishi Evolution	250000.00	2000
[2]	Proton Saga	60000.00	1800
[3]	Proton Wira	45000.00	1300
[4]	Toyota VIOS	85000.80	2000
[5]	BMW	500000.00	3500

### EXERCISE 3: Class

The following is a class declaration for `Book`, friend function, non-member function prototype and main program. The class consists of 4 data members and 3 function members. Answer all questions in this section.

```
1 // Program 2.12a  
2 class Book  
3 {  
4     // Declaration of 4 data members for class Book  
5     string bookName, bookAuthor, category;  
6     float bookPrice;  
7     public:  
8         //constructor  
9         Book() {bookName[]="xxx"; bookAuthor ="yyy";  
10                bookPrice = 0.0; category = "CS";}  
11  
12     // member function prototype declaration  
13     string checkAuthor() { return bookAuthor; }  
14     void giveDiscount(int);  
15     void printDetail(); // print all info about the book  
16     { cout << bookName << bookAuthor << bookPrice << category;  
17     }  
18  
19     // friend function declaration  
20     friend Book readValue();  
21  
22 }; // end of class declaration  
23  
24 // non-member function declarations  
25 void CheckBookInfo(Book[]);  
26  
27 main()  
28 { Book myBook[8];  
29   int discount;  
30   for (int i=0;i<8;i++)  
31       myBook[i] = readValue();  
32 }
```



```

33 cout << "Enter percentage amount of discount : ";
34 cin >> discount;
35 // Code to call function giveDiscount () using the array
36 for (int i=0;i<8;i++)
37     myBook[i].giveDiscount(discount);
38 // Statement that call function CheckBookInfo(Book[]) - 1e
39 } // end main()

```

- d. The following function in Program 2.12b is an application function for `readValue()`. Describe what the function does and what value does the function returns.

```

38 // Progrm 2.12b
39 Book readValue()
40 { Book aBook;
41   string bookName, bookAuthor, category;
42   float bookPrice;
43   cout << "Please enter the book title " << ": ";
44   getline(cin, aBook.bookName);
45
46   cout << "Please enter the name of the author " << ": ";
47   getline(cin, aBook.bookAuthor);
48
49   cout << "Please enter the book category " << ": ";
50   getline(cin, aBook.category);
51
52   cout << "Please enter the price of the book " << ": ";
53   cin >> aBook.bookPrice;
54   cin.ignore();
55
56   return aBook;
57 }

```

- e. Explain the purpose of statement 30 and 31 in `main()`.

```

for (int i=0;i<8;i++)
    myBook[i] = readValue();

```

- f. Write code for member function `giveDiscount(int)` that will set the new price for the book based on the percentage amount specified in the parameter. If the book was written by “Norbahiah” reduce the price based on the discount. No discount will be given to any book written by other authors. Give appropriate message if no discount is applied.
- g. Function `CheckBookInfo(Book[])` receives array of `Book`. Write implementation for the function that will count how many book is categorized as CS (Computer Science) book and use function `printDetail()` to print the detail information of the books in that category.
- h. Write a statement in `main()` that can call function `CheckBookInfo(Book[])`

**EXERCISE 4: Pass by Value**

Given the source codes in Program 2.13, answer all questions in this part.

```
1 //Program 2.13
2 class nombor
3 { int nom1, nom2;
4   public:
5     nombor (int, int);
6     void ubahnom (int, int);
7     get_nom1() {return nom1;}
8     get_nom2() {return nom2;}
9 };
10 nombor::nombor (int no1, int no2)
11 {nom1 = no2;
12   nom2 = no1;
13 }
14 void nombor::ubahnom (int ubahno1, int ubahno2)
15 {
16   nom1 += ubahno2;
17   nom2 += ubahno1;
18 }
19 void kira(nombor N)
20 {
21   if ((N.get_nom1() < 0) || (N.get_nom2() < 0))
22   {
23     N.ubahnom(5, 2);
24     cout << "\nNom 1 dalam fungsi = " << N.get_nom1();
25     cout << "\nNom 2 dalam fungsi = " << N.get_nom2();
26   }
27   else
28     cout << "\nTiada perubahan pada Nom 1 dan Nom 2";
29 }
30
31 void main()
32 {nombor uji (-3, 8);
33   cout << "\nNilai bagi Nom 1 = " << uji.get_nom1();
34   cout << "\nNilai bagi Nom 2 = " << uji.get_nom2();
35   kira (uji);
36   cout << "\nNilai bagi Nom 1 = " << uji.get_nom1();
37   cout << "\nNilai bagi Nom 2 = " << uji.get_nom2();
38 }
```

- Describe method of passing parameter to kira() function in Program 2.13.
- What type of parameter received by **kira()** function?
- Write the output of Program 2.13.
- Modify kira() function so that it will pass parameter by value.
- Give the output, if pass by value concept is implemented on kira() function.

**EXERCISE 5: DISTRUCTOR**

Give output of the following program.

```
1 // Program 2.14
2 #include <iostream.h>
3 int bil = 0;
4
5 class UJI
6 {private:
7     int nilai;
8 public:
9     UJI(int v)
10    { nilai = v;
11      cout << "\nCreating object with value: " << nilai ;
12    }
13    ~UJI()
14    { cout << "\nDestroying object with value :", nilai; }
15    void change()
16    { nilai *= nilai; }
17 };
18 void external()
19 {     UJI objek1(10), objek2(5);
20     objek1.change();
21 }
22 main()
23 {     test objek1(3), objek2(7);
24     objek2.change();
25     external();
26     return 0;
27 }
```

**EXERCISE 6: Header File and Implementation File**

C++ program that implements **abstract data type** divide the program into 3 files; header file, implementtion file and client file.

- a. Give brief discription for every types of files.
- b. Explain the advantages of dividing the program into the 3 files.