



**UNIVERSITI TUN HUSSEIN ONN MALAYSIA**

**FINAL EXAMINATION  
SEMESTER II  
SESSION 2012/2013**

**COURSE NAME** : OBJECT ORIENTED  
PROGRAMMING

**COURSE CODE** : BIT 20603

**PROGRAMME** : 2 BIT

**EXAMINATION DATE** : JUNE 2013

**DURATION** : 2 HOURS AND 30 MINUTES

**INSTRUCTION** : ANSWER ALL QUESTIONS

THIS QUESTION PAPER CONSISTS OF NINE (9) PAGES

**SECTION A**

Question **Q1- Q4** are based on the information given in **FIGURES Q1-Q3**.

**Q1** Name the class(es) involve in **FIGURES Q1-Q3**. (2 marks)

**Q2** For each class identified in **Q1**, name the methods belong to the class. (8 marks)

**Q3** Name the objects instantiated for every class. (2 marks)

**Q4** Assume the input sequence are as follows:

```
Betty
C++
100

Ramon
Pascal
78

Ahmad
C#
98
```

(a) Determine the output after executing the following program segment.

```
cout<<"A list of "<<size<<" students\n";
for (i=0; i<size; i++)
{
    newstudent.SetData();
    alist.X_Operation(newstudent);
}
alist.Z_Operation();
```

(10 marks)

(b) Determine the output after executing the following program segment.

```
cout <<"\nNext operation\n";
alist.Y_Operation();
cout <<"Finishing operation\n";
alist.Z_Operation();
cout <<"\nHappy Ending\n";
```

(10 marks)

```

//Filename: Student.cpp
#include <iostream.h>

class Student {
private:
    struct Data {
        char Name[25];
        char Course[30];
        int Result;
    } stdata;
public:
    void SetData();
    void GetData();
};

void Student::SetData()
{
    cout << "Enter student name: ";
    cin >> stdata.Name;
    cout << "Enter student course: ";
    cin >> stdata.Course;
    cout << "Enter student result: ";
    cin >> stdata.Result;};

void Student::GetData()
{
    cout << " \nStudent name : " << stdata.Name;
    cout << "\n";
    cout << " \nStudent course : " << stdata.Course;
    cout << "\n";
    cout << " \nStudent result : " << stdata.Result;
    cout << "\n";};

```

**FIGURE Q1**

```

//Filename: StdList.cpp
#include <iostream.h>
#include "Student.cpp"

class StudentList {
private:
    struct ListNode {
        Student astudent;
        ListNode *next;};
    ListNode *head;
public:
    StudentList();
    ~StudentList();
    int IsEmpty();
    void X_Operation(Student newstudent);
    void Y_Operation();
    void Z_Operation();
}; //class StudentList

StudentList::StudentList() {

```

```

        head = NULL; };

StudentList::~StudentList() {

    while (IsEmpty() != 0)
        Y_Operation();
    if (IsEmpty() == 0)
        cout << "\nEnding operation\n";};

int StudentList::IsEmpty() {
    if (head == NULL)
        return 0;
    else
        return 1;};

void StudentList::X_Operation(Student newstudent) {

    ListNode *newPtr = new ListNode;

    if (newPtr==NULL)
        cout<<"Cannot allocate memory";
    else
    {
        newPtr->astudent = newstudent;
        newPtr->next=head;
        head=newPtr;} };

void StudentList::Y_Operation() {
    if (IsEmpty()== 0)
        cout<<"\nBegin operation\n";
    else
    {
        ListNode *temp=head;
        head=head->next;
        temp->next=NULL;
        delete temp; } //end if};

void StudentList::Z_Operation() {
    ListNode *cur = head;
    if (IsEmpty()== 0)
        cout<<"\nFinishing point \n";
    else

        cout<<"Check this out : \n";
        while ( cur!=NULL){
            cur->astudent.GetData();
            cur = cur -> next; }
        cout<<"\n";        } };

```

**FIGURE Q2**

```

//Filename: StListDr.cpp
#include <iostream.h>
#include "StdList.cpp"

int main() {
const int size = 3;
int i;
    Student newstudent;
    StudentList alist;

```

```

cout<<"A list of "<<size<<" students\n";

    for (i=0; i<size; i++)
    {
        newstudent.SetData();
        alist.X_Operation(newstudent);
    }
    alist.Z_Operation();

    cout <<"\nNext operation\n";
    alist.Y_Operation() ;
    cout <<"Finishing operation\n";
    alist.Z_Operation();
    cout <<"\nHappy Ending\n";
    return 0;
};

```

### FIGURES Q3

Question **Q5-Q8** are based on **FIGURES Q4-Q6**.

- Q5** Name the classes as implemented in **FIGURES Q4, Q5** and **Q6**. (6 marks)
- Q6** Draw an appropriate diagram to show the relationship between the classes. (3 marks)
- Q7** Name the base class(es) in the implementation. (2 marks)
- Q8** Name the derived class(es) in the implementation. (2 marks)
- Q9** What are the methods can be invoked in the sub class? (10 marks)
- Q10** Implement a method for the class in **FIGURE Q6** that will receive two marks (integer) as parameters and return an average of two marks. (5 marks)

```
//Filename: Sum.cpp
#include <iostream.h>

class Sum {
public:
    int Sum2No(int no1, int no2);
    void DisplayDetails(int x, int y);
};

int Sum::Sum2No(int no1, int no2) {
    return no1 + no2;
};

void Sum::DisplayDetails(int x, int y) {
    cout << "\nThe sum of " << x << " and " << y;
    cout << " are " << Sum2No(x, y) << '\n';
};
```

**FIGURE Q4**

```
//Filename: Student.cpp

#include <iostream.h>
class Student {
private:
    char Name[25];
public:
    void SetName();
    void GetName();
}; // class Student

void Student::SetName()
{
    cout << "Enter student name: ";
    cin >> Name;
}; // method SetName

void Student::GetName()
{
    cout << " \nStudent name : " << Name;
    cout << "\n";
}; // method GetName
```

**FIGURE Q5**

```
//Filename: Test.cpp
#include <iostream.h>
#include "Sum.cpp"
#include "Student.cpp"

class Test: public Student, public Sum
{ private :
    int Result;
    public:
        Test();
        void SetResult();
        void GetResult();
};

Test::Test() {
Result = 0;
}

void Test::SetResult(){
int x, y;
cout << "Type the first mark: ";
cin >> x;
cout << "Type the second mark : ";
cin >> y;
Result = Sum2No(x,y);
};

void Test::GetResult() {
cout << "Result : " << Result ;
cout << "\n";
};
```

**FIGURE Q6**

**SECTION B****Q11** Refer to **FIGURE Q7**.

801109012345	Farish	12/07/2013	0830
810221115678	Alice	23/07/2013	0180
800705107896	Shae	20/08/2013	1800
820926048973	Farouq	12/10/2013	1900
801011016751	Black	11/07/2013	1520

**FIGURE Q7**

The information of bus passenger is stored in a file called *Ticket.dat* as shown in **FIGURE Q7**. Each passenger's information is as follows:

- i. IC Number
- ii. Passenger's Name
- iii. Departure Date
- iv. Departure Time

You are required to implement a class `Passenger` that contains passenger's information. Your class implementation should allow data to be read from the specified file and write the data on the screen.

Based on the above requirements, answer the following questions:

- (a) Draw a class diagram for the class `Passenger`. (8 marks)
- (b) Use `struct` mechanism to declare the attributes identified in **Q11(a)**. (12 marks)
- (c) Use C++ programming language to implement the following tasks.
- (i) Open the specified file to read the input. (2 marks)
  - (ii) Write a program segment that will read all the data in the file. (10 marks)



- (d) Implement the driver (main) that will instantiate an object of the class Passenger. Call the method from the class implemented in **Q11(c)**.

(8 marks)

-END OF QUESTION-