UTHM

Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
SEMESTER I
SESSION 2018/2019

COURSE NAME	: OBJECT ORIENTED PROGRAMMING
COURSE CODE	: BIC 20904
PROGRAMME CODE	: BIS / BIM / BIP / BIW
EXAMINATION DATE	: DECEMBER 2018 / JANUARY 2019
DURATION	: 3 HOURS
INSTRUCTION	: A) ANSWER **ALL** QUESTIONS
	B) PLEASE WRITE YOUR ANSWER IN THIS QUESTION BOOKLET

THIS QUESTION PAPER CONSISTS OF **ELEVEN (11)** PAGES

Q1 Answer Q1 based on **Figures Q1(a) – Q1(c).**

```
//Filename: Point.cpp
#include <iostream.h>
class Point{
      private: int x, y;
      public: int setValue(int a, int b, int c, int d);
              void getValue();
      protected:int p, q;};

int Point::setValue(int a, int b, int c, int d)
{x = a; y = b; p = c * 10; q = d * 100; return 1000; };

void Point::getValue(){
      cout <<"\n x = "<< x;
      cout <<"\n y = "<< y;
      cout <<"\n p = "<< p;
      cout <<"\n y = "<< y; };
```

Figure Q1(a)

```
//Filename: ThisShape.cpp
#include "Point.cpp"

class ThisShape:public Point {
      private: int result;
      public: void DisplayAnswer(); };

void ThisShape::DisplayAnswer(){
       result = setValue(10, 20, 30, 40);
       getValue();
       cout << "\nResult =" <<result;};
```

Figure Q1(b)

```
//Filename: Test.cpp
#include "ThisShape.cpp"

int main(){

 ThisShape test;
 test.DisplayAnswer();

 return 0;};
```

Figure Q1(c)

Determine whether each of the following statement is **VALID** or **INVALID**. Write your answer in the given column.

(10 marks)

Statement(s)	Answer
The relationship is known as single inheritance because there are three files implemented in the **Figures Q1(a) – Q1(c)**.	
In class `ThisShape`, the following statements can be implemented to display values in `p` and `q`: `cout << p << " and " << q;`	
The methods in class `Point` can be invoked in class `ThisShape` only.	
If `getValue()` is declared as protected, then calling the method `getValue()` in the derived class is not appropriate.	
The relationship between the classes is known as polymorphism.	

Q2 Answer **Q2(a) – Q2(d)** based on the information given in **Figure Q2.**

```
//Filename: Task.cpp

#include <iostream>
using namespace std;

class Task{
       double a = 1.0, b = 2.0;
    private:
       double *p1, *p2, p3 = 2.0;

   public:
       void calcTaskOne(double q1, double q2);
       void calcTaskTwo(double q3, double q4);
       void calcTaskThree(double q1,double q2,double q3,double q4);

   protected:
       double c, d;
       double calcTaskFour(double x, double y);
       double calcTaskFive(double x, double y);
};

void Task::calcTaskOne(double q1, double q2){
       p2 = &p3;
       *p2 = 4.0;
       a = p3 * q1;
       b = 10 * q2;
       cout << "\n Value of a = " << a << "\t and b = " << b;
       cout << "\n The answer for task one: " << calcTaskFive(a, b);
};

void Task::calcTaskTwo(double q3, double q4){
       p1 = &b;
       *p1 = a;
       c = a + q3;
       d = q3 + q4 + b;
       cout << "\n Value of c = " << c << "\t and d = " << d;
       cout << "\n The answer for task two: " << calcTaskFour(c, d);
            };

void Task::calcTaskThree(double q1, double q2, double q3, double q4){
       a = q1 + q3;
       b = a + q2;
       c = q4 + q3;
       d = q4;
       cout << "\n Value of a = " << a << "\t , b = " << b << "\t , c = " << c <<
"\t and d = " << d;
       cout << "\n The answer for task three: " << calcTaskFour(a, c);
};

double Task::calcTaskFour(double x, double y){ return x * y + 12; };
            double Task:: calcTaskFive(double x, double y){ return x * (y + 12);};
```

Figure Q2

(a) Draw a class diagram for **Figure Q2**.

(10 marks)

Answer:

(b) Determine the output of the following statements, implemented in main program.

```
#include <iostream.h>
#include "Task.cpp"
int main()
{
  Task task;

 task.calcTaskOne(2.0,4.0);
 task.calcTaskTwo(2.0,4.0);
  return 0;}
```

(14 marks)

Answer:

(c) Determine whether each of the following statements is **VALID** or **INVALID** to be implemented in a main program.

No	Statement(s)	Answer
1	`Task t[5];` `t[5].calcTaskOne(8.8,0.5);`	
2	`Task Test;` `Test.calcTaskFour(-1.5*2,-8.5/5);`	
3	`Task T[5];` `cout<<T[2].calcTaskFive(0.5,4.8);`	
4	`Task Num[5];` `Num[0].calcTaskTwo(0.5,4.8+10);`	

5	`Task *S1, S2;`
	
	`S2.calcTask(2.5,2.5);`
	`S1 = &S2;`	

(10 marks)

(d) State the difference between public, private and protected members.

(6 marks)

Answer:

Q3 (a) Explain the concept of :

(i) constructor
(ii) destructor

(4 marks)

Answer:

(b) Implement the constructors of the class in **Figure Q3(b)**.

```
class Sample
{
private:
    int x1,x2;
    double y;
public :
    Sample();
    Sample(int);
    Sample(int, int);
    Sample(int, double);
};
```

Figure Q3(b)

(10 marks)

Answer:

(c) Declare class Teacher shown in **Figure Q3(c)**. The `setName()` method shall allow user to input a teacher's name and the `getName()` method shall display the name of a teacher.

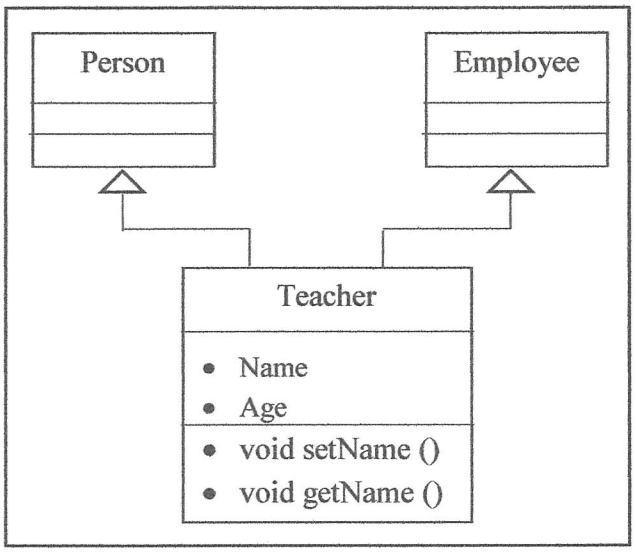

Figure Q3(c)

(11 marks)

Answer:

Q4 Answer **Q4(a) – Q4(b)** based on **Figures Q4(a)** and **Q4(b)**. The Product class consists of three attributes: ProductName, ProductId, and price. The SetProd() method is meant to obtain the input from users and GetProd() method is meant to display values in the attributes.

Product
- ProductName - ProductId - price
+ SetProd() : void + GetProd() : void

Figure Q4(a)

```
class ProductList  {
      private:
            struct ListNode {
                  Product aproduct;
                  ListNode *next;
            };
            ListNode *head;
      public:
            ProductList();
            int IsEmpty();
            void Add(Product newProduct);
            void Remove();
            void DisplayList();};

ProductList::ProductList() {
            head = NULL;  };

int ProductList::IsEmpty() {
            if (head == NULL)
                  return 0;
            else
                  return 1;
}; // method IsEmpty
```

(a) Declare the class for **Figure Q4(a)**. Apply `struct` definition to store product information shown in **Figure Q4(a)**.

(10 marks)

Answer:

(b) Implement method `Add(Product newProduct)` to add new product record at the beginning of a linked list.

(15 marks)

Answer:

- END OF QUESTIONS -