Universiti Tun Hussein Onn Malaysia

UNIVERSITI TUN HUSSEIN ONN MALAYSIA

FINAL EXAMINATION
(ONLINE)
SEMESTER I
SESSION 2020/2021

COURSE NAME	:	OBJECT ORIENTED PROGRAMMING
COURSE CODE	:	BIT 20603
PROGRAMME CODE	:	BIT
EXAMINATION DATE	:	JANUARY / FEBRUARY 2021
DURATION	:	2 HOURS AND 30 MINUTES
INSTRUCTIONS	:	1. ANSWER **ALL** QUESTIONS 2. STUDENTS SHOULD UPLOAD THE ANSWER BOOKLET (PDF/ WORD FORMAT) WITHIN 30 MINUTES AFTER EXAMINATION PERIOD

THIS QUESTION PAPER CONSISTS OF **SIXTH (6)** PAGES

Q1 Answer **Q1(a) – Q1(e)** based on the information given in **FIGURE Q1(a)** and **FIGURE Q1(b)**

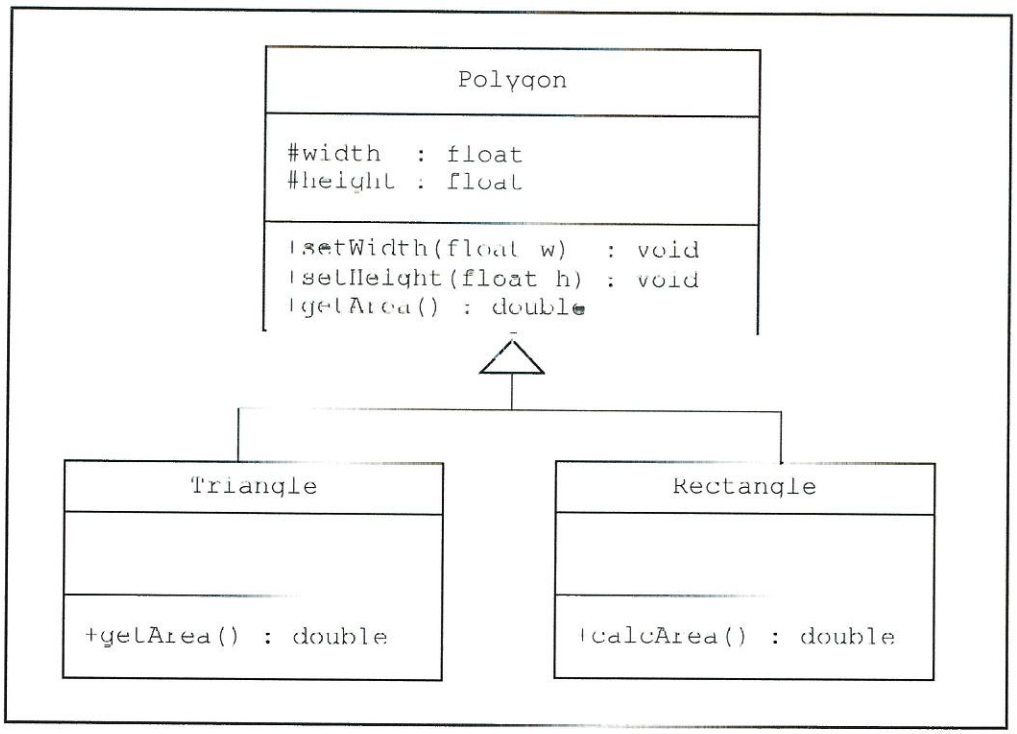

FIGURE Q1(a)

```
//Filename: driver.cpp
#include "Triangle.cpp"

int main(){
      Triangle t;
      t.setHeight(4);
      t.setWidth(6);
      cout<<t.getArea()<<endl;
}
```

FIGURE Q1(b)

(a) Indicate whether each of the following statement is **TRUE** or **FALSE**. Write your answer in the given space next to each statement

(10 marks)

Statements	Answer
Class `Shape` is the derived class of class `Triangle` and `Rectangle`.	
Public members of a base class can be access by other classes except derived classes	
Inheritance enhances the ability to reuse code.	
Method `getArea()` allows code sharing and code reuse between class `Polygon` and class `Triangle` only	
When objects instantiated from class `Triangle` invoke `getArea()` using dot operator, the function `getArea()` from class `Polygon` will be executed.	

(b) Assume that the driver file of the program is given as in **FIGURE Q1(b)**. What will happen during program compilation if the method in class `Polygon` is changed from `+setWidth(int w)` into `-setWidth(int w)`? Justify the reasons.

(6 marks)

(c) Write a C++ program to define class `Polygon` depicted in **FIGURE Q1(a)** with their attibutes and method implementations. Assume that the `getArea()` method of class `Polygon` is to be declared as `virtual` function. The method `getArea()` will return zero value.

(10 marks)

(d) Discuss the use of virtual functions in C++ and it relations with Polymorphism.

(8 marks)

(e) With appropriate examples, discuss **FOUR (4)** advantages of inheritance in Object Oriented Programming as compared to the traditional structured approach.

(10 marks)

Q2 Answer **Q2(a)** and **Q2(b)** based on **FIGURE Q2(a)**, **FIGURE Q2(b)** and **FIGURE Q2(c)**.

booklist.dat file

OPP	MyPublisher	20
C++	YourPublisher	34
Java	OurPublisher	17

FIGURE Q2(a)

Line	Codes
1.	//Filename: catalogue.cpp
2.	#include <iostream>
3.	
4.	using namespace std;
5.	
6.	class catalogue{
7.	
8.	struct detail{
9.	char title [50], publisher [50];
10.	int copy;} book;
11.	
12.	public:
13.	
14.	void readInput();
15.	
16.	};
17.	
18.	void catalogue::readInput(){
19.	
20.	input.open("booklist.dat");
21.	
22.	if(input.fail()){
23.	
24.	cout<<"unable to open filo\n";
25.	
26.	exit(1);
27.	
28.	}
29.	
30.	while(input>>book.title>>book.publisher>>book.copy){
31.	cout<<book.copy<<endl;
32.	cout<<book.title<<endl;
33.	cout<<book.publisher<<endl<<endl;}
34.	
35.	input.close();
36.	
37.	}
38.	

FIGURE Q2(b)

Line	Codes
1.	//Filename: catalogueD.cpp
2.	#include <iostream>
3.	#include "catalogue.cpp"
4.	
5.	int main(){
6.	catalogue library;
7.	cout<<"** List of books in FSKTM Mini Library **\n";
8.	library.readInput();
9.	}
10.	

FIGURE Q2(c)

(a) Fix the errors in the program `cataloque.cpp` in **FIGURE Q2(b)**. Identify its line code and provide with the correct answers.

(8 marks)

(b) Determine the output of the program.

(4 marks)

Q3 **Answer Q3(a) – Q3(c)** based on **FIGURE Q3(a)** and **FIGURE Q3(b)**.

Customer
- Name[30]:char - Id[10]:char - loyaltyPoint : float
+ SetCustomer():void + GetCustomer():void

FIGURE Q3(a)

```
class CustomerList  {
      private:
            struct ListNode {
                  Customer acustomer;
                  ListNode *next;
            };
            ListNode *head;
      public:
            CustomerList();
            int IsEmpty();
            void Add(Customer newCustomer);
            void Remove();
            void DisplayList();
};

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

FIGURE Q3(b)

(a) Write a class definition for **FIGURE Q3(a)**. Apply struct definition to store student's information shown in **FIGURE Q3(a)**.

(5 marks)

(b) Implement the constructor for the class `CustomerList` in **FIGURE Q3(b)**. In the constructor assign pointer `head` with null.

(3 marks)

(c) Implement method `Add(Customer newCustomer)` to add new instance at the end of the linked list for the class in **FIGURE Q3(b)**.

(16 marks)

END OF QUESTIONS