

**SECTION A – TRUE/FALSE QUESTIONS****[Total 10 marks]**

There are **FIVE (5)** questions in this section. For each statement, identify whether the statement is **TRUE/FALSE** and write your answer with your **REASON** in the given space. Each question carries **2** marks.

1. The following declaration and assignments are valid:

```
class Books
{
    public:
        Book()
        { cout << "constructor 1 called, ";}
        Book(double c, double u)
        { cout << "constructor 2 called, "; }
};
```

**Answer:**\_\_\_\_\_

**Reason:**\_\_\_\_\_

\_\_\_\_\_

2. Given the definition of a constructor of class **Circle** as follows.

```
Circle::Circle(int x=0, int y=0, int r=5)
{
    centerX = x;
    centerY = y;
    radius = r;
}
```

The following statements are valid declarations to create objects of the class.

```
Circle c1(10,10,20);
Circle c2;
```

**Answer:**\_\_\_\_\_

**Reason:**\_\_\_\_\_

\_\_\_\_\_

3. A **friend** function is not a member function; however, a **friend** function does have access to the **private** data members of the **class** in which it is included.

**Answer:**\_\_\_\_\_

**Reason:**\_\_\_\_\_

4. The following class declaration implies that the class **Bird** is a type of class **Animal**.

```
class Animal : public Bird
{
    .....
};
```

**Answer:**\_\_\_\_\_

**Reason:**\_\_\_\_\_

5. Static member functions can only access static member data.

**Answer:**\_\_\_\_\_

**Reason:**\_\_\_\_\_

**SECTION B – STRUCTURE QUESTIONS****[Total marks 60]**

There are **FOUR (4)** questions in this section. Answer all questions in the space provided. Each question carries **15** marks. The marks for each part of the question is as indicated.

**Question 1 (15 marks)**

- a) There are **10 lines** with syntax errors in the following program. Identify each of the errors and suggest the cause, and finally write the correct code in the table provided on the next page. [10 marks]

```

1  #include<iostream>
2  #include<string>
3  using namespace std;
4
5  class Student
6  {
7      private:
8          string name;
9      public:
10         void Student()
11             { name = ""; }
12
13         Student(string aName = "")
14             { name = aName; }
15
16         ~Student(string msg)
17             { cout << "The object has been destroyed" << endl; }
18
19         string getName() const;
20             { return name; }
21
22         void setName(string aName) const
23             { name = aName; }
24 };
25
26 int main()
27 {
28     student Student("Siti Aminah");
29     Student s1="Ramlee Puteh";
30     Student s2("Ahmad Mahmood");
31     Student s3;
32
33     cout << "Student's name: " << s2.name << endl;
34     s3.setName = "Mustafa Kamal";
35     s2.name = s1.name;
36
37     system ("pause");
38     return 0;
39 }
```



- b) Consider the following program which demonstrates the concept of friend functions used in calculating floor areas of different shaped rooms: square room and rectangular room. For this purpose two classes namely: **SquareRoom** and **RectangleRoom** are utilized. Complete the program by filling the blanks with appropriate statement(s) from guides in the comments or the formatting.

```
#include <iostream>
using namespace std;

class SquareRoom;

// i. for rooms with unequal sides // [0.25 mark]

class _____
{
    int width, height;

    public:
        RectangleRoom(int w = 1, int h = 1):width(w),height(h){}

        // ii. declare the prototype for function displayArea as a friend of
        // the class. It has no return value and accepts the address of
        // objects for RectangleRoom and SquareRoom respectively.

        _____ // [1 mark]
};

// iii. for rooms with equal sides // [0.25 mark]

class _____
{
    int side;

    public:
        SquareRoom (int s = 1):side(s){}

        // iv. declare the prototype for friend function displayArea again.

        _____ // [1 mark]
};
```

```
// v. Complete the definition of function displayArea

void _____ // [1 mark]
{
    cout << "The rectangular room area: "
        << _____ << endl; // [0.5 mark]
    cout << "The square room area: "
        << _____ << endl; // [0.5 mark]
}

int main () {
    _____ rec(5,10); // [0.25 mark]
    _____ sq(5); // [0.25 mark]

    displayArea(rec,sq);

    system ("pause");
    return 0;
}
```

**Question 2 (10 marks)**

a) Consider the following program. Fill in the blanks with the required statement(s).

```
#include <iostream>
using namespace std;

class Box
{ private:
    int x;
    int y;
    // i. declare a static integer variable z [0.5 mark]

    _____
```

```
public:
    Box()
    {   x=0,y=0,z++;   }
    // ii. Define a function putBox which accepts an integer value
    // to initialize static variable z [1.5 marks]

    _____

    _____

    _____

    void display()
    {   cout << x << " " << y << " " << z << endl;   }
};
// iii. Initialize static member z with 0 [1 mark]

    _____

int main()
{   // iv. Define Box object named one [0.5 mark]

    _____

    one.putBox(2);
    one.display();

    // v. Define Box object named two [0.5 mark]

    _____

    two.display();
    system("pause");
    return 0;
}
```

vi. What will be printed when the program is executed?

[1 mark]

- b) Given a class and a function named `SomeClass` and `someFunction()`, respectively, in the following program.

```
#include <iostream>
using namespace std;

class SomeClass
{
    private:
        int num;

    public:

        SomeClass(int i) {
            num = i;
            cout << "Constructor: " << num << endl;
        }

        SomeClass(const SomeClass &o) {
            num = o.num;
            cout << "Copy constructor: " << num << endl;
        }

        SomeClass& operator=(const SomeClass &o) {
            cout<< "Assignment operator: "<< o.num << "->" << num << endl;
            num = o.num;
            return *this;
        }

        ~SomeClass() {
            cout << "Destructor: " << num << endl;
        }
};

SomeClass someFunction(const SomeClass &a, SomeClass b, SomeClass *c) {
    return *c;
}
```

Write the output (into the provided table) that the program prints out when each of the statements below is executed. Assume the statements are written in the `main` function and executed sequentially. [5 Marks]

	Statment	Output [Answer: 0.5 each line]
i.	<code>SomeClass *a = new SomeClass(10);</code>	
ii.	<code>SomeClass b = *a;</code>	



iii.	<code>*a= b;</code>	
iv.	<code>delete a;</code>	
v	<code>someFunction(1, b, &amp;b);</code>	

**Question 3 (20 marks)**

a) Given a class diagram for the class **CellPhone** .

CellPhone
<code>-manufact : string</code> <code>-model : string</code> <code>retailPrice : double</code>
<code>+CellPhone ( );</code> <code>+CellPhone (man: string, mod: string, price: double);</code> <code>+setManufact(man: string): void</code> <code>+setModel(mod: string): void</code> <code>+setRetailPrice(price: double): void</code> <code>+getManufact(): string</code> <code>+getModel(): string</code> <code>+getRetailPrice(): double</code>

Assume that class **CellPhone** has been defined in **CellPhone.h**. In your main program:

- i. Create an array of three **CellPhone** objects that initialize the following data to the appropriate constructor.

[3.5 marks]

Manufacturer	Model	Retail Price
Samsung	Galaxy SIII Mini	899.00
Nokia	Lumia 4	1089.00
Sony	Xperia 8	1499.00

- ii. Use a for loop to display the data for each cell phone on the screen.

[3.5 marks]

- b) Given three classes named **Shape**, **Rectangle** and **Square** respectively, and some objects of these classes as shown in the following program.

```
1 #include<iostream>
2 using namespace std;
3
4 class Shape
5 {
6     private:
7         int positionX;
8
9     public:
10        int positionY;
11
12        Shape()
13        {
14            positionX = 0;
15            positionY = 0;
16
17            cout << "A shape has been created and "
18                << "positioned at the origin." << endl;
19        }
20
21        Shape(int x, int y)
22        {
23            positionX = x;
24            positionY = y;
25
26            cout << "A shape has been created at position "
27                << "(" <<positionX << ", " <<positionY << ")"
28                << endl;
29        }
30
31        void methodX()
32        { }
33
34 }; // class Shape
35
36
37 class Rectangle : protected Shape
38 {
39     private:
40         int width;
41
42     protected:
43         int length;
44
45     public:
46         Rectangle():Shape(10,10)
47         {
48             width = 0;
49             length = 0;
50             cout << "An empty rectangle has been positioned "
51                 << "at (10,10)" << endl;
52         }
53 }
```

```

54
55         Rectangle(int w, int l)
56         {
57             width  = w;
58             length = l;
59             cout << "A rectangle with length "
60                 << length << " and width " << width << endl;
61         }
62
63         void methodY()
64         { }
65
66 }; // class Rectangle
67
68
69 class Square : private Rectangle
70 {
71     public:
72         Square(int size):Rectangle(size,size)
73         {
74             cout << "A square has been created." <<endl;
75         }
76
77         void methodZ()
78         { }
79
80 }; // class Square
81
82
83
84
85 int main()
86 {
87     Shape shape(100,200);
88
89     Rectangle rect;
90
91     Square square(10);
92
93     system("pause");
94     return 0;
95 }

```

- i. What will the **lines 87–91** print onto the screen when the program runs? [3 marks]

Line	Statement	Output [0.5m for each line of output]
87	Shape shape(100,200);	

89	Rectangle rect;	
91	Square square(10);	

- ii. Based on the objects created in the main function of the program (lines 87–91), determine the member variables that each object owns. Write (**Own**) in the corresponding blank cells below if the object own or has a copy of the variable and write (**No**) if otherwise.

[4 marks]

	Member Variables			
Object	positionX	positionY	length	width
shape				
rect				
square				

- iii. With reference to the same member variables in (i), determine whether they are accessible inside the methods and by the object listed below. Write (**Accessible**) in the corresponding blank cells below if it is accessible and write (**No**) if otherwise.

[6 marks]

	Member Variables			
Method/Object	positionX	positionY	length	width
methodX( )				
methodY( )				
methodZ( )				
shape				

**Question 4 (15 marks)**

Given an incomplete program with two classes named **Point** and **Circle**, respectively, as shown below. Complete the program based on the tasks to accomplish in questions (a) to (l) stated in the program.

[15 marks]

```
#include<iostream>
#include<string>
#include<cmath>

using namespace std;

class Point
{
    private:
        int x, y;

    public:

        void print() const
        {   cout << "(x=" << x << ", y=" << y << ")" << endl; }

        Point(int a=0, int b=0)
        {   x = a;
            y = b;
        }
}
```

// (a). Declare and define an overloaded operator for the minus operator (-).  
// The operator returns the distance between two points. The distance is calculated as:

//  $distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

// where the two points are  $(x_1, y_1)$  and  $(x_2, y_2)$  respectively.

[1.5 marks]

// **(b).** Declare and define an overloaded operator for the plus operator (+).  
// The operator adds the coordinates  $x$  and  $y$  of two points and returns the  
// new point. For example if the two points are **(1,2)** and **(10,20)**, respectively, then  
// the resultant point will be **(11,22)**. [1.5 marks]



**};** // class Point

**class Circle**

**{**

private:

Point \*center;  
int radius;

public:

// **(c).** Declare and define a default constructor for the class.  
// The constructor allocates memory for the center point, and set the point  
// to **(0,0)**. The radius is also set to **0**. [1 mark]



// **(d).** Declare and define another constructor that sets the center point and radius of the circle from arguments. [1.5 marks]

```
Circle(Point aCenterPoint, int aRadius)
{

}

}
```

// **(e).** Declare and define the destructor for the class. [1 mark]  
// The destructor frees the memory allocated for the center point.

//**(f).** Declare and define a method that shifts or moves the center point of a circle at particular distance in both direction *x* and *y*.  
// For example, if the existing center point is **(10,20)** and the distance to shift is **(3,-5)**, then the center point becomes **(13,15)**. [1 mark]

```
void shift(Point distance)
{

}

}
```

// **(g).** Declare and define a method to print the information of center point and radius of the corresponding circle. Determine and specify whether the method should be read-only. [1 mark]

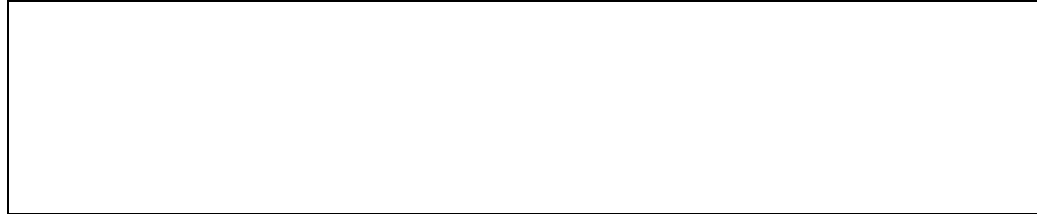
```
void print() const
{

}

}
```



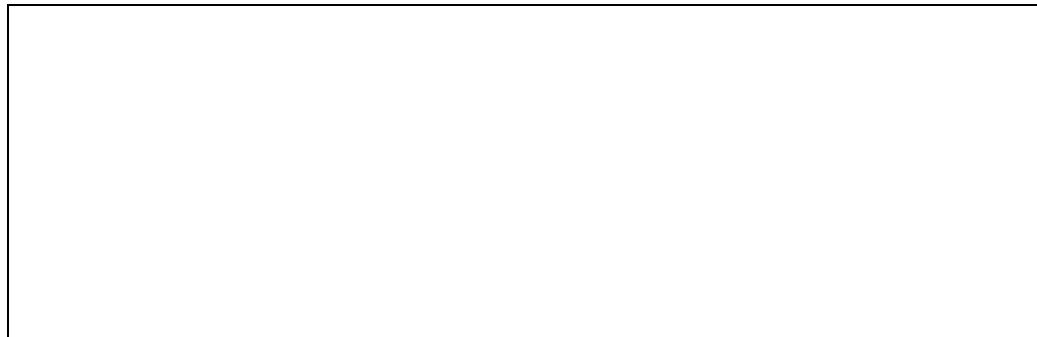
// **(h).** Declare and define a method that returns the center point of a circle.  
// The return type should be of type **Point**. Determine and specify whether  
// the method should be read-only method.. [1 mark]



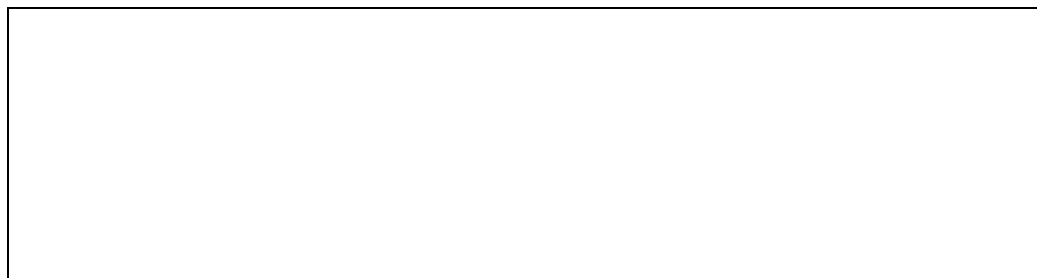
**};** // class Circle

**int main()**  
**{**

// **(i).** Declare and initialize an array to hold 4 objects of Circle.  
// The center points of the circles are **(0,0)**, **(20,20)**, **(50,50)** and **(100,100)**, respectively,  
// whereas their radius are **0, 5, 10** and **40**, respectively. [1.5 marks]

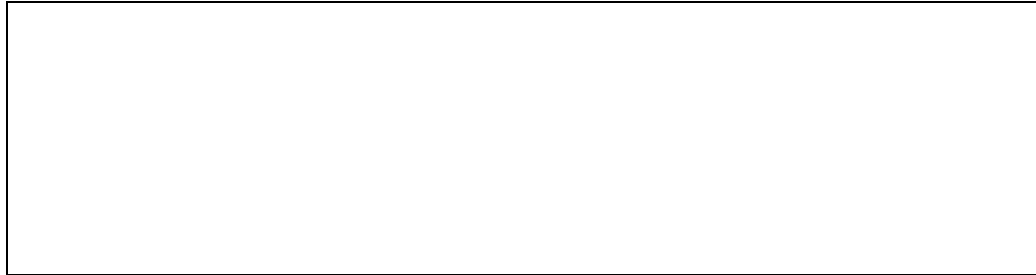


// **(j).** Using an appropriate loop, shifts the center points of all the circles with 10 units for  
// both directions x and y. [1.5 marks]



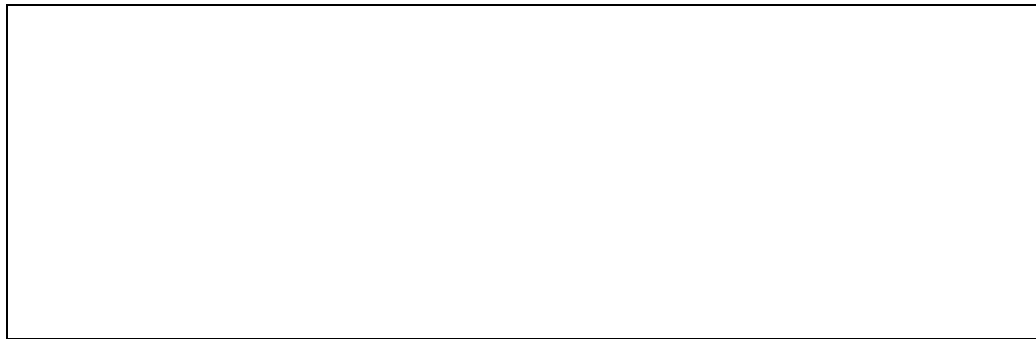
// **(k).** Using another loop, print the information of center point and  
// radius of all the circles. .

[1 mark]



// **(l).** Calculate and print the distance between the first and last circles.  
// The distance is calculated from the centers of those two circles.

[1.5 marks]



```
system ("pause");  
return 0;
```

```
}
```

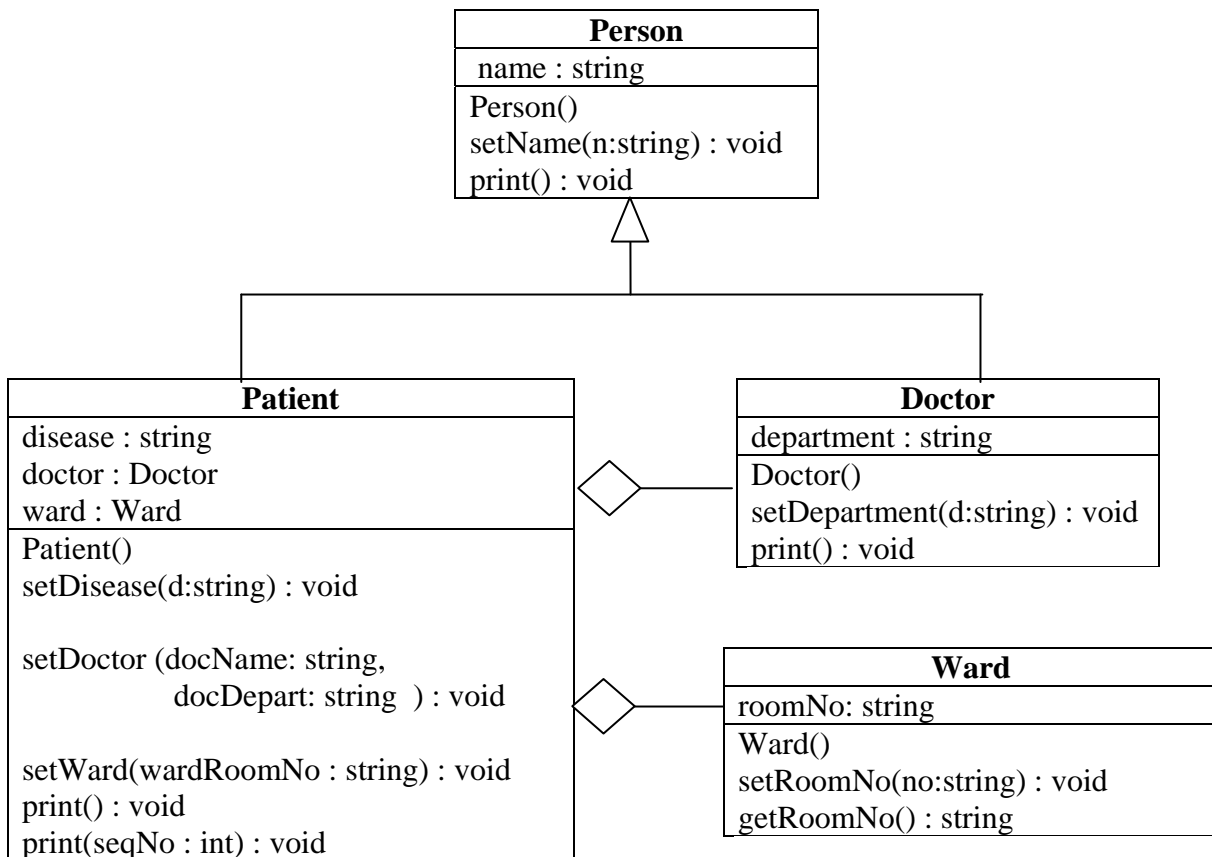
**SECTION C - PROGRAMMING QUESTION****[Total marks 30]**

This section consists of **ONE (1)** question only.

**Question**

An IT company has been awarded a project by a local hospital to develop a simple version of patient management system. The purpose of the system is to record patients that are hospitalized in the hospital. The records include the information of the patient, the doctor in-charged for the patient, and the ward that the patient is staying.

You are a member of a software development team in the company and your team has been assigned to run the project. Your team leader who is also the system analyst for the project has done the requirement analysis and came out with the following design and specifications.

**UML Class Diagram:**

**Specifications:**

The following specifications describe the detail of the above design and other requirements to comply with.

*Program*

- The program needs to be written in C++ language and implements the object-oriented programming concept of aggregations and inheritances.

*Class Implementation*

- Implement all the classes using inline approach.

*Member Variables*

- Specify the member variable of the base class so that it is accessible by the derived classes.
- Specify the member variables of all the classes so that they are inaccessible by the objects of the classes.

*Member Functions*

- Implement all the constructors of the classes such that they initialize the member variables of the classes with empty strings.
- Determine and specify each of the member functions of the classes whether they should be read-only.
- Implement the concept of redefining and overloading for the method **print** where appropriate. The following table describes the output that should be produced by each of the methods **print**

- 

Method <b>print</b>	Description	Example Output
of class <b>Person</b>	Do nothing	
of class <b>Doctor</b>	Print the doctor's name and department	Doctor in-charged ===== Doctor's Name : Dr. Abu Hassan Department : Intensive Care Unit

of class <b>Patient</b> <ul style="list-style-type: none"> <li>The redefined method <b>print</b></li> </ul>	Print the patient's name, disease and ward room number.	Patient's Name : Ali Bin Ahmad Disease : Lung Cancer Ward No : B101
<ul style="list-style-type: none"> <li>The overloaded method <b>print</b></li> </ul>	Print the same patient's information above, together with a sequence number. This method should call to the redefined method above. The sequence number is taken from the argument of the method.	Patient No. 1 Patient's Name : Ali Bin Ahmad Disease : Lung Cancer Ward No : B101

### The main function

- Declare an array to hold a list of patients.
- Using an appropriate loop, fill in each patient in the array with related data entered from the keyboard. The user firstly needs to enter the number of patients.
- Using another loop, list the information of all the patients.
- The following figures illustrate examples of user input and the corresponding output when the program runs. Note that the **bold texts** indicate input from the user.

### User Input

```

How many patients to enter? => 2

Enter the following information:
  Patient's Name => Ali Bakar
  Disease       => Lung Cancer
  Ward No      => G890
  Doctor's Name => Dr. Ramli Samsudin
  Department   => Intensive Care Unit

Enter the following information:
  Patient's Name => Siti Aminah Salleh
  Disease       => Critical Injury
  Ward No      => C403
  Doctor's Name => Dr. Kamariah Binti Razali
  Department   => Emergency Room

```

*Program Output*

```
Patient #1
Patient's Name : Ali Bakar
Disease       :
Ward No:      : G890
Doctor In-charged
=====
Doctor's Name: Dr. Ramli Samsudin
Department   : Intensive Care Unit

Patient #2
Patient's Name : Siti Aminah Salleh
Disease       : Critical Injury
Ward No:      : C403
Doctor In-charged
=====
Doctor's Name: Dr. Kamariah Binti Razali
Department   : Emergency Room
```

As a programmer in the team, write a C++ program that implements the design and complies with the specifications above.

***[This page is intentionally left blank for answering Section C.]***

***[This page is intentionally left blank for answering Section C.]***



***[This page is intentionally left blank for answering Section C.]***