SECTION A: MULTIPLE CHOICES SINGLE ANSWER [30 Marks]

Instruction: Read each question below carefully and choose the letter A, B, C, or D that **BEST** describes the answer. Write your answer in the answer booklet. Each question carries **1.5 marks**.

1. In the given scenario, the class `Library` has a collection of books, but the books can exist independently and can be associated with multiple libraries.
   What type of relationship exists between the class `Library` and the class `Book` ?
   A. Inheritance
   B. Compositions
   C. Aggregation
   D. Encapsulation

2. Determine the relationship/s is/are implemented in the following code segment.

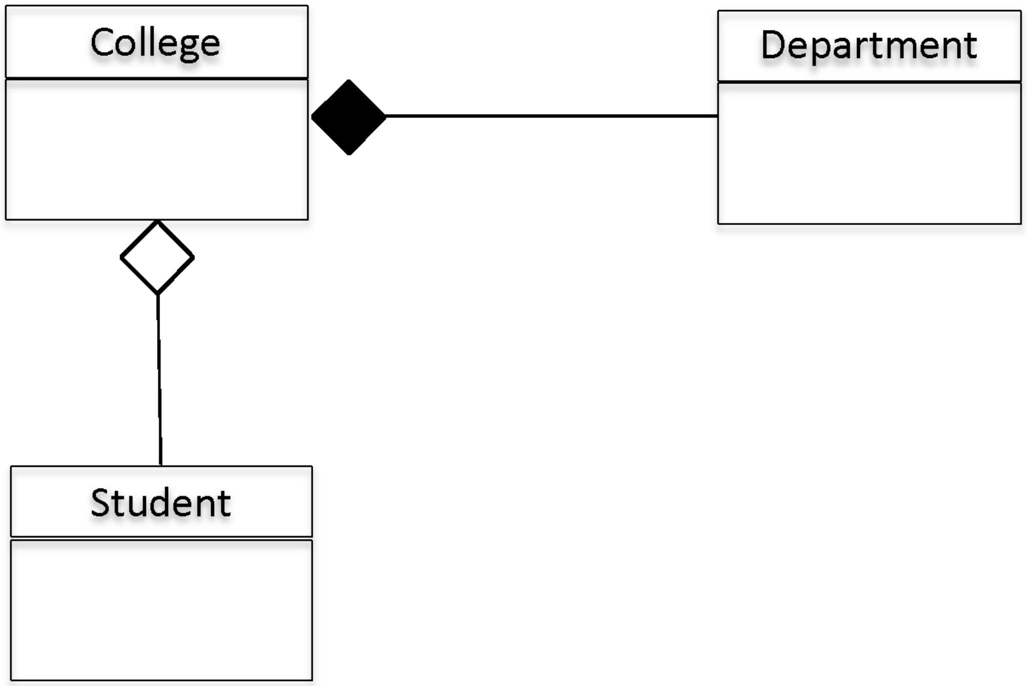
```
class Department {
    // details codes are left out for brevity
};

class College {
    // details codes are left out for brevity
};

class Student {
    // details codes are left out for brevity
    Department dept;
    College *college;
};
```

   A. Aggregation and composition
   B. Aggregation only
   C. Composition only
   D. No relationship

3. Based on the following class diagram, which type of relationship/s is/are implemented in the diagram?



   A. Aggregation and composition
   B. Association and aggregation
   C. Association and composition
   D. Composition only

4. Which of the following is **FALSE** about the association?
   A. Association indicates the relationships between classes through their objects.
   B. The association can be one-to-one and many-to-many relationships only.
   C. The association can be one-to-one and one-to-many relationships.
   D. Association allows one object instance to cause another to perform and then act on its behalf.

```
 1 #include <iostream>
 2 #include <string>
 3 using namespace std;
 4 
 5 class Car {
 6     private:
 7         string make, model;
 8     public:
 9         Car(string make, string model) {
10             this->make = make;
11             this->model = model; }
12         void print(){ cout << make << "-" << model; } };
13 
14 class Person {
15     private:
16         Car* car;
17         string name;
18         int year;
19     public:
20         Person(string name, int year) {
21             this->name = name;
22             this->year = year; }
23         void buyCar(Car* c){ car = c; }
24         void sellCar(){ car = NULL; }
25         void print(){
26             cout << name << "-" << year << ": ";
27             if (car!= NULL) car->print();
28             cout << endl; } };
29 
30 int main() {
31     Person person1("Jack", 2023);
32     Car *car1 = new Car("Proton", "X70");
33     person1.buyCar(car1);
34     person1.print();
35     person1.sellCar();
36     person1.print();
37     return 0; }
```

5. Consider the given program above. Which of the following statements is **FALSE** about the program ?

   A. Line 16 implements the aggregation concept.

   B. Line 16 implements the composition concept.

   C. A `person` may have a `car`.

   D. An object of `Person` may be created without an object of `Car`.

6. Determine the result of the following program?

```
using namespace std;

class Base1 {
    public:
        Base1()
        { cout << " Base1" << endl;  }
};

class Base2 {
    public:
        Base2()
        { cout << "Base2" << endl;  }
};

class Derived: public Base1, public Base2 {
    public:
        Derived()
        { cout << "Derived" << endl;  }
};

int main() {
    Derived d;
    return 0;
}
```

A. A compilation error occurs.

B. Base1 Base2 Derived

C. Base2 Base1 Derived

D. A run-time error occurs.

7. If a base class is inherited with protected access mode, which of the following statements is **TRUE**?

A. Public and Protected members of the base class become protected members of the derived class.

B. Only protected members become protected members of the derived class.

C. Only private members of the base class become private of the derived class.

D. All private, protected, and public members of the base class become private of the derived class.

```
 1  #include <iostream>
 2  using namespace std;
 3  
 4  class Animal {
 5      public:
 6          Animal()
 7          { cout << "Animal constructor executing" << endl;
 8  }
 9  
10          ~Animal()
11          { cout << "Animal destructor executing." << endl;
12  }
13   };
14  
15  class Cat : public Animal {
16      public:
17          Cat() : Animal()
18          { cout << "Cat constructor executing" << endl; }
19  
20          ~Cat()
21          { cout << "Cat destructor executing" << endl; }
22   };
23  
24  int main(){
25      Animal *myAnimal = new Cat();
26  
27      delete myAnimal;
28  
        return 0;
    }
```

8. Consider the program above. Which of the following statements is **TRUE** about the program?

   A. In Line 25, both constructors in class `Animal` and `Cat` will execute.

   B. In Line 27, both destructors in class `Animal` and `Cat` will execute.

   C. In Line 25, only constructors in class `Animal` will execute.

   D. In Line 27, only destructors in class `Cat` will execute.

9. What is the purpose of Inheritance of object-oriented programming?

   A. To establish a "has a" relationship between classes.

   B. To define a relationship between parent and child classes.

   C. To encapsulate data and methods within a class.

   D. To create an instance of a class.

```
Class Base{
   private: x;
   protected: y;
   public: z;
};

Class Derived: private
Base{};
```

10. Consider the class declaration above. Which of the following statements is **TRUE** about how the inherited base class members appear in the class `Derived`?

   A. x is inaccessible, y is private, and z is protected.
   B. x is inaccessible, y is private, and z is private.
   C. x is inaccessible, y is protected, and z is public.
   D. x is protected, y is public, and z is public.

11. Which of the following is the **CORRECT** syntax to declare a Pure Virtual Function?
   A. `virtual void display()=0;`
   B. `virtual void display ()==0;`
   C. `void virtual display ()==0;`
   D. `void display virtual ()=0;`

12. Which one of the following statements is TRUE about a pure virtual method?
   A. A pure virtual method is a method in a parent class declared as virtual without any definition
   B. A pure virtual method is a method in a parent class declared as virtual with at least one definition
   C. A pure virtual method is a method in a child class declared as virtual with at least one definition
   D. A pure virtual method is used to implement the polymorphism concept.

13. Identify abstract class/es in the following program.

```
class Person{
    string name;
    public:
        Person(string n):name(n){}
        virtual void print()const = 0;
};

class Lecturer : public Person{
    string department;
    public:
        Lecturer(string d):department(d){}

};

class Student : public Person{
    string matric;
    public:
        Student(string m):matric(m){}
        void print() const {cout << matric;}

};

class PostGraduate : public Student{
    string research;
    public:
        Student(string r):research(r){}
        virtual void setProject() = 0;

};
```

I - Person

II - Lecturer

III - PostGraduate

IV - Student

A. Only I

B. I and II

C. I, II, and III

D. I, II, III, and IV.

14. Which of the following concepts **DOES NOT APPLY** to the following program?

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

class Calculation{
private:
    int num1, num2;
public:
    Calculation():num1(0), num2(0) {}
    Calculation(int n1, int n2):num1(n1), num2(n2) {}
    int getNum1() { return num1; }
    int getNum2() { return num2; }
    virtual void display() {
         cout << num1 << "&" << num2 << ": "; }
};

class Addition : public Calculation {
private:
    int sum;
public:
    Addition() { sum = 0; }
    Addition(Calculation *c) {
         sum = c->getNum1() + c->getNum2(); }
    void display() { cout << sum << "; "; }
};

class Multiplication : public Calculation{
private:
    int product;
public:
    Multiplication() { product = 0; }
    Multiplication(Calculation *c) {
         product = c->getNum1() * c->getNum2(); }
    void display() { cout << product; }
};

int main(){
    Calculation* p = new Calculation(10,23);
    Addition* a = new Addition(p);
    Multiplication* m =new Multiplication(p);
    Calculation* s[3] = {p, a, m};
    for(int i = 0; i < 3; i++) { s[i]->display(); }
    return 0;
}
```

A. Virtual method

B. Abstract class

C. Dynamic binding

D. Overriden method

15. Consider the program below. Which part of the program implements the concept of polymorphism?

```
Class Shape {
    public:
    virtual string getPerimeter(){return 0;}
    int getType() {return 0;}
};
Class Rectangle : public Shape {
    public:
        double getPerimeter() { return 2 (l + w);}
        int getType() {return 1;}

};


class Circle : public Shape {
    public:
        void getPerimeter() {return 2 * 3.142 * r; }
        int getType() {return 2;}
};
```

A. Class `Shape`

B. Class `Rectangle and` Class `Circle`

C. Method `getPerimeter()`

D. Method `getType()`

16. Based on the program below, which of the following statements is **TRUE** about the `try` and `catch`?

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

int main(){
    try  {
        throw 'a';
    }
    catch (int x)  {
        cout << "Caught ";
    }
    return 0;
}
```

A. The code in the try-block is called an exception handler.

B. The code in the try-block is called protected code.

C. The code in the catch-block is called protected code.

D. If no catch block that matches the exception is found, the program will keep on executing up to infinite.

17. Which of the following statements is **TRUE** about the program below?

```
 1 #include <iostream>
 2 #include <string>
 3 using namespace std;
 4
 5 template <typename T>
 6
 7 class Data{
 8 private:
 9     T data;
10 public:
11     Data(T value) : data(value) {}
12     void print() { cout << data << " "; }
13 };
14
15 int main() {
16     Data<const char*> c("good luck");
17     Data<int> t(2);
18     Data<string> s("final exam");
19     c.print();
20     t.print();
21     s.print();
22     return 0;
23 }
```

A. The output of this program is as follows: `g 2 final exam`.

B. `T` is the type of `Data` template that replaces char, int, and string.

C. The `template` is the keyword to influence the data type.

D. In Line 9, `T` is the class, and the `data` is the object to relate to the template.

18. Choose a statement that is **TRUE** about error handling with exceptions.

A. Error handling is done by allowing the error to occur then it is handled by catching it later.

B. Error handling is done by anticipating an error before it happens.

C. Error handling is done by safeguarding code with the if block to prevent any error to occur.

D. Error handling is done by ignoring the execution of code that may raise errors.

19. What is the output for the following program if the input value is **`120 120.55`**?

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

template <class T>
T dividetwo(T number) {
    return number/2;
}

int main() {
    int ival;
    double dval;

    cout << setprecision(3);
    cout << "Enter an integer and a double value: ";
    cin  >> ival >> dval;
    cout << "Value divided by two: ";
    cout << dividetwo(ival) << " and "
         << dividetwo(dval) << endl;
    return 0;
}
```

A. `Value divided by two: 60 and 60.3`

B. `Value divided by two: 60 and 60.0`

C. `Value divided by two: 60.3 and 60.3`

D. `Value divided by two: 60.0 and 60.0`

20. Which of the following statements is **TRUE** about vectors and arrays?

A. Vectors have a dynamic size whereas arrays have a static size.

B. Both vectors and arrays have a dynamic size.

C. Both vectors and arrays have a static size.

D. Vectors have a static size whereas arrays have a dynamic size.

SECTION B: STRUCTURED QUESTIONS [70 Marks]

Instruction: This section consists of THREE (3) questions. Answer all questions. The marks are as indicated in the question.

Question 1 **(30 Marks)**

a. Describe the difference between aggregation and composition.
Provide examples for each type of relationship.

(6 marks)

b. Discuss the following programming concepts and for each of them, give an example use case or scenario where the concept is applicable.

i inheritance

ii polymorphism

iii abstract class

iv class template

(20 marks)

c. Explain with an appropriate example, how map is useful in C++ programming.

(4 marks)

Question 2 **(20 Marks)**

Consider the following program.

```
class Engine {
protected:
    double horsepower;
};

class Wheel {
protected:
    double size;
};

Class Radio {
protected:
    string brand;
};

class Vehicle {
protected:
    string manufacturer;
    Wheel wheel;
};

class Bicycle : public Vehicle {
protected:
    bool hasSidecar;
};

class SportsCar : public Vehicle, public Engine {
protected:
    bool isConvertible;
};
```

a. Based on the program, determine whether the following classes form any relationship in terms of Object-Oriented Programming. If so, write the name of the relationship, otherwise write "No relationship".

   i. `Vehicle` and `Bicycle`

   ii. `Wheel` and `Vehicle`

   iii. `Vehicle` and `Engine`

   iv. `Bicycle` and `Sportscar`

   v. `Sportscar` and `Vehicle`

(10 marks)

b. Draw the class diagram for the program.

(10 marks)

Question 3 **(20 Marks)**

Answer the following questions based on the given program.

```
#include<iostream>
#include<string.h>
#include<stdio.h>
using namespace std;

class publisher {
    char pname[15];
    char hoffice[15];
    char address[25];
    double turnover;

protected:
    char phone[3][1O];
    void register();

public:
    publisher();
    ~publisher();
    void enter data();
    void disp data();
};

class branch {
    char bcity[15];
    char baddress[25];

protected:
    int no_of_emp;

public:
    char bphone[2][10];
    branch();
    ~branch();
    void have data();
    void give data();
};

class author : public branch, publisher {
    int aut_code;
    char aname[2O];
    float income;

public:
    author();
    ~author();
    voidgetdata();
    voidputdata();
};
```

a. Which type of inheritance is implemented in the program?

(1 mark)

b. Name the base class(/es) and derived class (/es).

(3 marks)

c. Give the sequence of Constructor/Destructor Invocation when object of class author is created.

(6 marks)

d. Which data members are accessible from the objects of class `author`.

(5 marks)

e. Which member functions are accessible from the object of class `author`.

(5 marks)