Mid Paper Assignment

Submitted by : Hasnain Abbas

Sap Id : 42917

Course : OOP – 2A

Section : BSCS – 3

Submitted to : Sir Shehzad Shameer

Q.No 1

Write the outputs of the following cod snippets .

(a)

```
class base{
        private:
            int a;
        protected:
                int b;
        public:
                void fun(){
                        cout<<"this works";


                }


};
class child:private base{
        public:
                void fun1(){
                        base::fun();
                        base::b=0;
                }
};
int main()
{
   child c1;
  // c1.fun();
  c1.fun1();
}
```

Output :

No output because the syntax error in this cod.

(b)

```
class TableofContents{
        private:
                list<string>items;
                public:
              TableofContents(){
                cout<<"Table of Content is shown\n";
                 }
        void addItem(string item){
                item.push_back(items);
        }
};
class Book{
        public:
                TableofContents toc;
                list<string>sections;
                list<string>chapters;
                Book(){
                        toc=TableofContents();
                }
};
int main()
{
  Book book1=Book();
  getchar();
  return 0;

}
```

<u>Output :</u>

```
                Table of Content is shown
                 Table of Content is shown
```

<u>{c}</u>

```
class A{
        int a;
        public:
                A(int i){
                        a=i;
                }
                void assign(int i){
                        a=i;
                }
                int return_value(){
                        return a;
                }
};
int main()
{
  A obj;
  obj.assign(5);
  cout<<obj.return_value();
}
```

Output :

No output because the error in this cod .

Constructor is missing in this cod .

(d)

```
class A{
        int a;
        public:
                A(){
                        cout<<"A's constructor called"<<endl;
                }
};
class B{
        static A a;
        public:
                B(){
```

```
                cout<<"B's constructor called";
            }
            static A get(){
                return a;
            }
};
 A B::a;
int main()
{
   B b;
   A a1=b.get();
   A a2=b.get();
   A a3=b.get();
}
```

Output :

A's constructor called one time because static calls one time .

```
A's constructor called
```

Q.No 2

Give the short answer to the following questions.

(1)

What is the purpose of access modifiers in oop languages?

Answer:

Access modifiers in object-oriented programming (OOP) languages are used to control the visibility and accessibility of class members (i.e., fields, methods, constructors) from outside the class. The purpose of access modifiers is to provide encapsulation, which is a fundamental principle of OOP.

There are generally three types of access modifiers in OOP languages:

1. Public: A public member is accessible from any part of the program, including other classes and packages.

2. Private: A private member is only accessible within the same class in which it is declared. It cannot be accessed from outside the class, including from derived classes.

3. Protected: A protected member is accessible within the same class and its subclasses (i.e., derived classes). It cannot be accessed from outside the class hierarchy.

Access modifiers help to enforce information hiding, which means that the internal details of a class are hidden from other classes, and can only be accessed through well-defined interfaces. This allows classes to be modified and improved without affecting other parts of the program, and helps to prevent bugs and errors that can arise from unintended modifications to internal class details. Additionally, access modifiers can help to improve the readability and maintainability of code by making it clear which parts of a class are intended to be used by other classes, and which are internal implementation details.

If we want to access the private member of a class in a child class what do we need to change ?

Answer:

If you want to access a private member of a class in a child class, you need to change the access modifier of that member to either protected or public.

In OOP, private members are only accessible within the same class, so they cannot be directly accessed in a child class. However, protected members are accessible within the same class and its subclasses, which means that child classes can access protected members of their parent class. Public members, on the other hand, are accessible from anywhere, including child classes.

To change the access modifier of a member in a class, you simply need to modify the access specifier in the class definition. For example, if you have a private field called "myPrivateField" in a class called "MyClass", you can change its access modifier to protected by modifying the field declaration as follows:

```java
public class MyClass {
    protected int myPrivateField;
    // other class members
}
```

Alternatively, you could change the access modifier to public:

```java
public class MyClass {
    public int myPrivateField;
    // other class members
}
```

Note that changing the access modifier of a private member may have implications for the encapsulation and information hiding of your class, so it's important to consider the design implications carefully before making such a change.

(3)

Determines the accessibility of functions and data members in the following scenarios ?

(a)

A private data member is declared in a class in accessible by its object in the main function .

(No its not Accessible)

(b)

A protected function defined in parent class by the functions of the child class .

(Yes its Accessible)

(c)

A public data member of the parent class by the object of the child class .

(Yes its Accessible)

Q.No 3

There are Five (5) Errors (Syntax and Logical) in cod given below . Identify error lines and write coreect cod .

```
class B1{
        public:
                i; // The error in line #3 - int
                                int j;
                void g(int){}
};
class B2{
        public:
                int j;
                void g(){}
};
class D:public B1;class public B2{ // The error in line #12 - class D:public B1,public B2
        public:
                int i;
};
int main(){
        D dobj;
        D *dptr=&dobj; // The error in line #18 -* dptr=&dobj;
        dptr->i=5;
        dptr->j=10;
        dobj.g(); // The error in line #21 - d.g();
}
```

Q.No 4

you have to develop a game that has multiple characters .
these characters share some common properties like id,name,
maximum power and strength.there are other properties as
well that they have their own like Doremon has properties

like a list of name gadgets and the name of partner , Benton
has the watch name , a list of name powers and total charge
of the watch . There ara also common actions that they can
perform like walk , jump and eat. Doremon can show gadgets ,
launch attack and fly. Benton can perform the actions like
rotate watch , fight and drive.Implement the game using inheritance
in c++?

Answer

I can define a base class called "Character" that contains the
common
properties and actions of all characters. Then, we can create
derived
classes for each specific character, which inherit from the
"Character"
class and add their own unique properties and actions.

Cod

```
#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Character {
public:
    int id;
    string name;
    int max_power;
    int strength;

    void walk() {
        cout << name << " is walking.\n";
    }

    void jump() {
        cout << name << " is jumping.\n";
    }

    void eat() {
        cout << name << " is eating.\n";
    }
};

class Doremon : public Character {
public:
    vector<string> gadgets;
    string partner_name;

    void show_gadgets() {
        cout << "Gadgets of " << name << ": ";
        for (string gadget : gadgets) {
            cout << gadget << ", ";
        }
        cout << endl;
    }

    void launch_attack() {
        cout << name << " is launching an attack!\n";
    }

    void fly() {
        cout << name << " is flying.\n";
    }
};

class Benton : public Character {
public:
    string watch_name;
    vector<string> powers;
    int watch_charge;

    void rotate_watch() {
        cout << name << " is rotating the watch.\n";
    }

    void fight() {
        cout << name << " is fighting.\n";
    }

    void drive() {
        cout << name << " is driving.\n";
    }
};

int main() {
    Doremon d;
    d.id = 1;
    d.name = "Doremon";
    d.max_power = 100;
    d.strength = 50;
    d.gadgets = {"Take-copter", "Anywhere Door", "Bamboo-
Copter"};
```

```
d.partner_name = "Nobita";

d.walk();
d.jump();
d.eat();
d.show_gadgets();
d.launch_attack();
d.fly();

Benton b;
b.id = 2;
b.name = "Benton";
b.max_power = 150;
b.strength = 80;
b.watch_name = "Omnitrix";
b.powers = {"Heatblast", "Four Arms", "XLR8"};
b.watch_charge = 75;

b.walk();
```

```
    b.jump();
    b.eat();
    b.rotate_watch();
    b.fight();
    b.drive();

    return 0;
}
```

End

Thank You So Much