ASSIGNMENT- 5(PPL)

NAME: SOURABH PATEL

ADMISSION NO: U19CS082

1. **Given the following class hierarchy, which inherited memberscan be accessed without qualification from within the VMI class?Which requires qualification? Explain your reasoning.**

```
 struct Base {
 void bar(int); // public by default
protected:
int ival; };

struct Derived1 : virtual public Base {
 void bar(char); // public by default
void foo(char);
 protected:
char cval;
};

struct Derived2 : virtual public Base {
 void foo(int); // public by default
protected:
int ival;
char cval;
 };
 class VMI : public Derived1, public Derived2 { };
```

   ⇨ The inherited members bar and ival can be accessed unrestricted from withinthe VMI class: bar exists in both the shared base class Base and the derivedclass Derived1, but the priority of the specific derived class instanceis higher than the shared base class instance, so in the VMI class Ifyouaccess bar without restriction, you will access the bar instanceinDerived1. Ival exists in both the shared base class Base and the derivedclass Derived2. Similarly, in the VMI class, you can access ival withoutrestriction, and you can access the ival instance in Derived2. The inherited members foo and cval need to be restricted: both existinDerived1 and Derived2. Both Derived1 and Derived2 are derived classesofBase and have the same access priority. Therefore, if you access fooandcval without restriction in the VMI class, then There will be ambiguity.

2)**Given the following class hierarchy:**

```
class Class { ... };

 class Base : public Class { ... };

 class D1 : virtual public Base { ... };
```

class D2 : virtual public Base { ... };

class MI : public D1, public D2 { ... };

class Final : public MI, public Class { ... };

(a) In what order are constructors and destructors run on a Finalobject?

⇨

Class(); //run by Base class default constructor

Base(); //D1 & D2 virtual base class are initialised first

D1(); //D1 & D2 are indirect non virtual base classes

D2();

MI();

Class();

Final(); //most derived class

Virtual base classes are initialized first. In this case, Baseisa virtual base class. However, since Base is derived directly from Class, it's defaultconstructor will run Class() before it does its work.

(b) **A Final object has how many Base parts? How many Class parts?**
1 Base subparts and 2 Class subparts

(c) **Which of the following assignments is a compile-time error?**
Base *pb; Class *pc; MI *pmi; D2 *pd2;
(a) pb = new Class;
Error: invalid base to derived type conversion
(b) pc = new Final;
Error: Class is inaccessible directly due to ambiguity
(c) pmi = pb;
Error: invalid base to derived type conversion
(d) pd2 = pmi;
Valid, but pmi is uninitialized

3)**Given the following classes, explain each print function:**

```
class base {
public: string name() {
 return basename; }
 virtual void print(ostream &os) {
 os << basename; }
private: string basename; };
class derived : public base {
public:
void print(ostream &os) {
print(os); os << ""<<i; }
private:
```

int i;
};
If there is a problem in this code, how would you fix it?
Void print(ostream &os) {
base::print(os); os <<<i;}

(4)Given the classes from the previous problem and the followingobjects, determine which function is called at run time:
base bobj; base *bp1 = &bobj; base &br1 = bobj; derived dobj; base *bp2 = &dobj; base &br2 = dobj;
(a) bobj.print(); (b) dobj.print(); (c) bp1->name(); (d) bp2->name(); (e) br1.print(); (f) br2.print();
(e) br1.print(); & (f) br2.print(); are called at run time.