Heaven's Light is Our Guide Rajshahi University of Engineering & Technology Department of Computer Science & Engineering

Lab Manual

Course Code: **CSE 1204 (Sec A)**Course Title: Sessional based on CSE 1203

Module 3 [Inheritance]: (for Week 3)

Problem Statement: You have the create an inheritance among **Father**-->**Son** -->**GrandSon** class. The **father** class has the following data members

```
class Father{
   private:
      int money;
   protected:
      int gold;
   public:
      int land;
};
```

Now write the **Son** and **GrandSon** classes with **private/protected/public** access modifier and do the following:

- i) Try to access **money**, **gold** and **land** from Son class
- ii) Try to access **money**, **gold** and **land** from GrandSon class
- iii) Find the values of money, gold and land when different access modifer is used in the following table

Class		In Son class			In GrandSon class		
Son	GrandSon	money	gold	land	money	gold	land
public	public	?	?	?	?	?	?
protected	public	?	?	?	?	?	?
private	public	?	?	?	?	?	?
public	protected	?	?	?	?	?	?
protected	protected	?	?	?	?	?	?
private	protected	?	?	?	?	?	?
public	private	?	?	?	?	?	?
protected	private	?	?	?	?	?	?
private	private	?	?	?	?	?	?

Topic 2 [**Types of Inheritance**]: Learn and Test different types of inheritance in C++. In each inheritance draw the class diagram with class chain and try to access the data members of bases classes from child classes.

i) Single inheritance

```
class B:public A{
                                      int main(){
class A{
                 //write public
  private:
                                       B b;
    int x;
               method to //access
                                      //call methods of
                                      class B
protected:
               x,y & z
                                      return 0;
  int y;
public:
                                      }
  int z;
```

ii) Multi-level inheritance

iii					
class A{	class	B:public A{	class C:public	<pre>int main(){</pre>	
private:	}		B{	C c;	
int x;			//write public	//call	
protected:			//method to	//methods of	
int y;			//access x,y & z	//class C	
<pre>public:</pre>			}	return 0	
int z;				}	
}					

iii) Multiple inheritance

	iii) Watapie iiiiertanee					
class A{ class B{		class C:public A,	<pre>int main(){</pre>			
	private:	private:	Public B{	C c;		
	int x;	int p;	//write public	//call		
	protected:	protected:	method //to access	//methods of		
	int y;	int q;	//x,y,z,p,q & r	//class C		
	<pre>public:</pre>	public:	}	return 0		
	int z;	int r;		}		
	}	}				

iv) Heirarchical inheritance

```
class C:public A
                                                   int main(){
class A{
            class B:public A
                                                    B b;
  private:
            //write public
                                  //write method
    int x;
                                                    C c;
            method to access
                                                    //call
protected:
                                public to access
                                x,y & z
                                                   //methods of
  int y;
            x,y & z
public:
                                }
                                                   //class B & C
              }
  int z;
                                                    return 0
```

v) Hybrid (Diamond) inheritance [virtual class]

```
class B:public A class C:public A
class A{
                                               class D:public
                                                               int main(){
 private:
                             {
                                               B, public C {
                                                                Dd;
   int x;
                                                //write
                                                               //call
protected:
                             }
                                               public method
                                                               //methods of
             }
  int y;
                                               to access x,y
                                                               //class D
                                                               return 0
public:
                                               & z
  int z;
                                               }
                                                               }
```

Topic 3 [Constructor & Destructor in inheritance]: Write the constructors & destructors for different types of inheritance are given as follows. Also follow and write the sequence of their execution.

i) Single inheritance

```
class B:public A{
class A{
                                                int main(){
                       private:
                                                 B b:
 private:
                         int bx;
                                                //call methods of
    int ax;
 public:
                      public:
                                                class B
 //write
                    //write constructor to
                                                return 0;
  constructor to
                    initialize bx
                                                 }
  initialize ax
                    //Write method to sum ax
  //Write
                    and bx
  destructor
                    //Write destructor
```

ii) Multi-level inheritance

```
private:
                  class B:public A {
                                       class C:public B {
                                                            int main(){
                     private:
   int ax;
                                         private:
                                                             C c;
public:
                        int bx;
                                             int cx;
                                                            //call
                     public:
                                                            //methods of
 //write
                                          public:
                      //write
                                                            //class C
 constructor to
                                           //write
                                           constructor to
                                                            return 0
 initialize ax
                      constructor to
 //Write
                      initialize bx
                                           initialize cx
                                       //Write method to
 destructor
                      //Write
}
                      destructor
                                       sum ax, bx and cx
                                       //Write destructor
```

iii) Multiple inheritance

private:	class B{	class C:public A,	<pre>int main(){</pre>
int ax;	private:	Public B{	C c;
<pre>public:</pre>	int bx;	private:	//call
//write	<pre>public:</pre>	int cx;	//methods of
constructor	//write	public:	//class C
to initialize	constructor	//write	return 0
ax	to	constructor to	}
//Write	initialize	initialize cx	
destructor	bx	//Write method to sum	
}	//Write	ax, bx and cx	
	destructor	//Write destructor	
	}	}	

iv) Heirarchical inheritance

```
int main(){
                 class B:public A
                                    class C:public A {
class A{
                                                          B b;
 private:
                                      private:
                                                          C c;
                                          int cx;
    int ax;
                  private:
 public:
                       int bx;
                                      public:
                                                         //call
//write
                   public:
                                       //write
                                                         //methods of
                                                         //class B & C
constructor to
                     //write
                                        constructor to
initialize ax
                     constructor
                                        initialize cx
                                                         return 0
//Write
                     to initialize
                                    //Write method to
                                                         }
                                    sum ax, bx and cx
destructor
                                    //Write destructor
 }
                     //Write
                     destructor
```

v) Hybrid (Diamond) inheritance [virtual class]

class A{	class B:public	class C:public A	class D:public	<pre>int main(){</pre>
private:	A {	{	B, public C {	D d;
int ax;	private:	private:	private:	//call
public:	int bx;	int cx;	int dx;	//methods of
//write	public:	<pre>public:</pre>	<pre>public:</pre>	//class D
constructor	//write	//write	//write	return 0
to initialize	constructor to initialize	constructor	constructor	}
ax	bx	to	to initialize	
//Write	//Write	initialize	dx	
destructor	destructor	cx	//Write	
}		//Write	method to	
1	}	destructor	sum ax, bx	
			cx and dx	
		}	//Write	
			destructor	
			}	