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

Lab Manual

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

Module 5 [Inheritance]: (for Week 5)

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

ii) **Multi-level inheritance**

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

iii) **Multiple inheritance**

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

iv) **Heirarchical inheritance**

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

v)   Hybrid (Diamond) inheritance [virtual class]

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

iii) Multiple inheritance

	class B{	class C:public A, Public B{	int main(){

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

iv) Heirarchical inheritance

class A{	class B:public A {	class C:public A {	int main(){

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

v) Hybrid (Diamond) inheritance [virtual class]

class A{	class B:public A {	class C:public A {	class D:public B, public C {	int main(){

private:
  int ax;
 public:
//write
 constructor
 to
 initialize
 ax
//Write
 destructor
}
	private:
	  int bx;
	 public:
	//write
	 constructor
	 to initialize
	 bx
	//Write
	 destructor
	
	 }
		private:
		   int cx;
		  public:
		  //write
		  constructor
		  to
		  initialize
		  cx
		//Write
		destructor
		
		}
			private:
			    int dx;
			   public:
			//write
			constructor
			to initialize
			dx
			//Write
			method to
			sum ax, bx
			cx and dx
			//Write
			destructor
			}
				D d;
				//call
				//methods of
				//class D
				return 0
				}