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 | ? ⊀ | ? 🗸 | ? 🗸 | . X ¿ | ? ~ / | 5∕ |
| protected | public | ? 🗙 | ? 🗸 | ? / | 5Χ | ?✓′ | .5 - \ |
| private | public | ? * | ? ~ | · ?< | ? 1 | ? X | .× |
| public | protected | 3 ⊀ | ? / | ? / | ? × | ? 🏏 | ?/ |
| protected | protected | ? ′ X | ?/ | ? / | ? 🔀 | ? / | ?/ |
| private | protected | ? ⊀ | ? 🗸 | <u>-</u> -ذ | 5 ⊀ | ? 🔀 | 7 |
| public | private | ? X | ? / | ? / | ? X | ? / | ?/ |
| 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.

⇒ Single inheritance

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

Multi-level inheritance

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

ள்) Multiple inheritance

| 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 | | | |
| <pre>int y;</pre> | int q; | //x,y,z,p,q & r | //class C | | | |
| <pre>public:</pre> | <pre>public:</pre> | } | return 0 | | | |
| int z; | int r; | | } | | | |
| } | } | | | | | |

iw Heirarchical inheritance

```
int main(){
class A{
            class B:public A
                                class C: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
                                                   //class B & C
public:
                                }
              }
  int z;
                                                   return 0
```

<u>v</u>

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.

Single inheritance

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

ii) Multi-level inheritance

```
class B:public A {
                                       class C:public B {
                                                            int main(){
private:
   int ax;
                     private:
                                         private:
                                                             C c;
public:
                        int bx;
                                             int cx;
                                                            //call
                                                            //methods of
 //write
                     public:
                                          public:
                                                            //class C
 constructor to
                      //write
                                           //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

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

Heirarchical inheritance

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

Hybrid (Diamond) inheritance [virtual class]

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