ITEC243 – Lecture Session (04-MAY-2020)

Inheritance (is-a) Relationship:

Syntax of creating inheritance relationship between the mother class and the child class

class <name of the derived class> : access specification <mother class>

where access specification is either private, protected, or public

Are constructors inherited by the derived classes??? NO

```
class checkpoint{
  private:
        int a;   //inaccessible!!!!!
  protected:
        int b, c;
        void setA(int a)
        {       this->a=a;          }
  public:
        checkpoint( ) //constructor
        {       this->a=0; this->b=0; this->c=0;        }
        void setB(int b)
        {       this->b=B;          }
        void setC(int c)
        {       this->c=c;          }
};
```

```
class quiz{
private:
        int d;
public:
       quiz( )
       { this->d=0; }
        void setD(int d)
        {  this->d=d;  }
};
```

Question 1) List private, protected and public data members of derived class

class quiz : private checkpoint

private: d, b, c, setA( ), setC( ), setB( ), ~~checkpoint( )~~

protected: none

public: quiz( ), setD( )

2) **class quiz : protected checkpoint**

private: d

protected: b, c, setA( ), setB( ), setC( )

public: quiz( ), setD( )

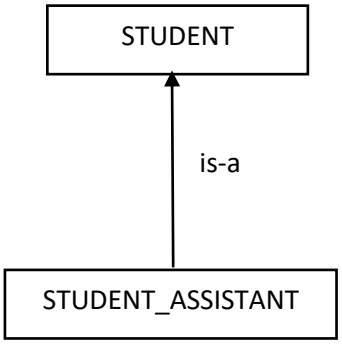
3) **class quiz : public checkpoint**

private: d

protected: b, c, setA( )

public:setB( ), setC( ), setD( ), quiz( )

Example: Single Inheritance Relationship



```
//student.h
class student{
protected:
	long stdid;
	string name;
public:
	student()
	{
		cout << "Enter student id and name:";
		cin >> this->stdid;
		getline(cin, this->name);
	}
	long getStdid()
	{
		return this->stdid;
	}
	string getName()
	{
		return this->name;
	}
	void setstdId(long stdid)
	{
		this->stdid = stdid;
	}
	void setName(string name)
	{
		this->name = name;
	}
};
```

Solution 1: Let's create PRIVATE inheritance relationship between "student" and "student assistant" classes.

```
//stdassistant.h
class std_assistant :private student
{
private:
	float range, working_hour;
public:
	std_assistant()
	{
		cout << this->stdid << " " << this->name << endl;
		cout << "Enter range and the working hours for above student:";
		cin >> this->range >> this->working_hour;
	}
	float getRange()
	{
```

```
		return this->range;
	}
	float getWorkinghours()
	{
		return this->working_hour;
	}
	float getWeeklySalary()
	{
		return (this->range*this->working_hour) * 5;
	}
	void printInfo()
	{
		cout << getStdid() << "\t" << getName() <<"\t";
	}
};
#include<iostream>
#include<string>
using namespace std;
#include"student.h"
#include"stda.h"
void main()
{
	student stdobj1;
	std_assistant saobj1;
	cout << stdobj1.getStdid() << "\t" << stdobj1.getName() << endl;
	saobj1.printInfo();
	cout << "Salary:"<<saobj1.getWeeklySalary() << endl;
	system("pause");
}
```

Solution 2: Let's create PROTECTED inheritance relationship between "student" and "student assistant" classes.

```
//sta2.h
class std_assistant :protected student
{
private:
	float range, working_hour;
public:
	std_assistant()
	{
		cout << this->stdid << " " << this->name << endl;
		cout << "Enter range and the working hours for above student:";
		cin >> this->range >> this->working_hour;
	}
	float getRange()
	{
```

```
		return this->range;
	}
	float getWorkinghours()
	{
		return this->working_hour;
	}
	float getWeeklySalary()
	{
		return (this->range*this->working_hour) * 5;
	}
	void printInfo()
	{
		cout << getStdid() << "\t" << getName() << "\t";
	}
};
//stds2.cpp
#include<iostream>
#include<string>
using namespace std;
#include"student.h"
#include"stda2.h"
void main()
{
	student stdobj1;
	std_assistant saobj1;
	cout << stdobj1.getStdid() << "\t" << stdobj1.getName() << endl;
	saobj1.printInfo();
	cout << "Salary:" << saobj1.getWeeklySalary() << endl;
	system("pause");
}
```

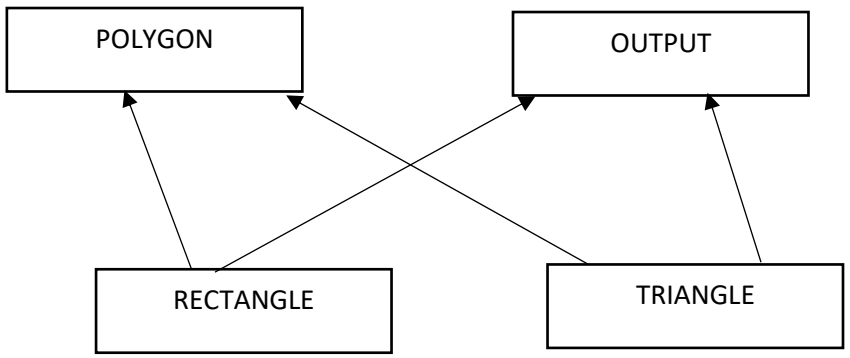
Solution 3: Let's create PUBLIC inheritance relationship between "student" and "student assistant" classes.

```
//stda3.h
class std_assistant :public student
{
private:
	float range, working_hour;
public:
	std_assistant()
	{
		cout << this->stdid << " " << this->name << endl;
		cout << "Enter range and the working hours for above student:";
		cin >> this->range >> this->working_hour;
	}
	float getRange()
```

```
	{
		return this->range;
	}
	float getWorkinghours()
	{
		return this->working_hour;
	}
	float getWeeklySalary()
	{
		return (this->range*this->working_hour) * 5;
	}
};

//stds3.cpp
#include<iostream>
#include<string>
using namespace std;
#include"student.h"
#include"stda3.h"
void main()
{
	student stdobj1;
	std_assistant saobj1;
	cout << "Student\n";
	cout << stdobj1.getStdid() << "\t" << stdobj1.getName() << endl;
	cout << "Student Assistant\n";
	cout << saobj1.getStdid() << "\t" << saobj1.getName() << "\t";
	cout << "Salary:" << saobj1.getWeeklySalary() << endl;
	system("pause");
}
```

Multiple Inheritance Relationship



Solution:

```
//polygon.h
class polygon{
protected:
	int width, height;
public:
	polygon(int width, int height)
	{
		this->width = width;
		this->height = height;
	}
};
//output.h
class output{
public:
	void show(int i)
	{
		cout << i << "\n";
	}
};
//rectangle.h
class rectangle :public polygon,  public output
{
public:
	rectangle(int w, int h) :polygon(w, h)
	{	}
	int getarea()
	{
		return this->width*this->height;
	}
};
//triangle.h
class triangle :public polygon, public output
{
public:
	triangle(int w, int h) :polygon(w, h)
	{	}
	int getarea()
	{
		return (this->width*this->height) / 2;
	}
};
//main
#include<iostream>
using namespace std;
#include"polygon.h"
#include"output.h"
#include"rectangle.h"
#include"triangle.h"
void main()
{
	rectangle robj(4, 5);
```

```
        triangle tobj(8, 6);
        robj.show(robj.getarea());
        tobj.show(tobj.getarea());

        system("pause");
}
```