Constructor in Inheritance

Default constructor of Base class

- Default Constructor of Base class will be called when object of Derived class is created.
- If Base class is having nay user-defined constructor then it must also have user-defined default constructor

Calling Parametrised constructor of Base class

It should be called from constructor of Derived class
Example:

```
class Cuboid : public Rectangle
{
        private :
        int height;
        public:
        Cuboid(int lint b, int h); // Base class constructor should not be called from prototype
};
Cuboid::Cuboid(int lint b, int h) : Rectangle(l , b) // calling base class constructor
{
        height=h;
}
```

Private member of Base in Derived class Constructor

No. Private members of Base class are not accessible in derived class. Private members of Base class are inherited to Derived class, but not accessible.

Copy Constructor by reference

- Copy constructor must take parameter by reference. If it is taking parameter by value then it has to create an object and it will class constructor again.
- It may become recursive call to the constructor. Constructor calling constructor.

Private Constructor

Yes constructor can be declared as private, but we cannot create the object directly. It can be done using static functions

Example:

```
class Test
{
	int x,y;
	Test(int a, int b)
	{
		x=a;
		y=b;
	}
	public:
	static Test * CreateObject() // this function will create an object.
	{
		Test t=new Test(10,10);
		return t;
	}
};
int main()
{
	Test *t=Test::CeateObject();
}
```

Virtual Base class

I multiple-path inheritance, a derived class may get the duplicate features via multiple parent classes. To avoid duplicacy we make parent class as virtual