Classification

- Animal
  - Mammal
    - Rodent
      - Mouse
    - Primate
    - Squirrel
  - Reptile
    - Cats
    - Rabbit
Enjoy a variety of personal banking options from First American. The following outlines a number of First American products. If you have any questions, please visit any First American Branch or contact us.

**Select your Personal Banking Solution:**

**Checking**
- Bank a lot without spending a lot: ValueFirst® Checking.
- Few checks and prefer PC Banking or ATMs: Select Access.
- Earn interest on checking dollars: First Interest Checking
- You are 55 years or better: 55 & Better Silver
- Premium checking features with higher interest rates than a personal checking account: First American Platinum Checking
- Write less than 10 checks per month, or bank through an ATM: Budget Checking.
- Younger than 24 years old and in school: Student Checking
- Less than 20 transactions per month (excluding ATM, POS, and CheckCard): First Account
- Make the most out of every dollar: Tailored Money Sweep
Inheritance

A class which is a subtype of a more general class is said to be inherited from it.

The sub-class inherits the base class’ data members and member functions.
Inheritance cont’d

- A sub-class has all data members of its base-class plus its own
- A sub-class has all member functions of its base class (with changes) plus its own
- Inheritance is meant to implement sub-typing (don’t abuse it)
Inheritance

- is-a relationship
Classification

Polygon

Rectangle

Triangle
```cpp
#include <iostream>
#include <iostream>
using namespace std;
using namespace std;

class Polygon {
    protected:
        int width, height;
    public:
        void set_values (int a, int b) {
            width=a; height=b;
        }
};

int main () {
    CRectangle rect;
    CTriangle trgl;
    rect.set_values(4,5);
    trgl.set_values(4,5);
    cout << rect.area() << endl;
    cout << trgl.area() << endl;
    return 0;
}
```
#include <iostream>
#include <iostream>

using namespace std; using namespace std;

class Polygon {
    protected:
        int width, height;
    
    public:
        void set_values (int a, int b) {
            width=a; height=b;
        }
};

class Rectangle: public Polygon {
    
    public:
        int area () {
            return (width * height);
        }
};

int main () {
    CRectangle rect;
    CTriangle trgl;
    rect.set_values(4,5);
    trgl.set_values(4,5);
    cout << rect.area() << endl;
    cout << trgl.area() << endl;
    return 0;
}
#include <iostream>

using namespace std;

class Polygon {
    protected:
        int width, height;
    public:
        void set_values (int a, int b) {
            width=a; height=b;
        }
};

class Rectangle: public Polygon {
    public:
        int area () {
            return (width * height);
        }
};

class Triangle: public Polygon {
    public:
        int area () {
            return (width * height / 2);
        }
};

int main () {
#include <iostream>
using namespace std;

class Polygon {
    protected:
        int width, height;
    public:
        void set_values (int a, int b) {
            width=a; height=b;
        }
};

class Rectangle: public Polygon {
    public:
        int area () {
            return (width * height);
        }
};

class Triangle: public Polygon {
    public:
        int area () {
            return (width * height / 2);
        }
};

int main () {
    Rectangle rect;
    Triangle trgl;
    rect.set_values (4,5);
    trgl.set_values (4,5);
    cout << rect.area() << endl;
    cout << trgl.area() << endl;
    return 0;
}