Polymorphism

- Code is said to be polymorphic if executing the code with different types of data produces different behavior
- Program in the general, rather than program in the specific
- Virtual functions make polymorphism possible

Consider

```
#include <iostream>
using namespace std;
class Def1
{
  public:
    Def1() {cout << "Def1 Constructor" << endl;}
    ~Def1 () {cout << "Def1 Destructor" << endl;}
    void Foo () {cout << "Def1 Foo" << endl;}
};

class Def2 : public Def1
{
  public:
    Def2 () {cout << "Def2 Constructor" << endl;}
    ~Def2 () {cout << "Def2 Destructor" << endl;}
    void Foo () {cout << "Def2 Foo" << endl;}
};
```

What is the output? Why?

```
int main ()
{
     Def1 *pcDef1_1 = new Def1;
     Def1 *pcDef1_2 = new Def2;
     pcDef1_1->Foo();
     pcDef1_2->Foo();
     delete pcDef1_2;
     delete pcDef1_1;
}
```


Polymorphism

- You can tell the compiler to select the more specialized version of a member function by declaring the member function to be a virtual function
- Declare a virtual function by prefixing its declaration with the word virtual

What is the output? Why?

- If the following 2 changes are made to the previous program, what is the output? Why?

```
virtual void Foo () {cout << "Def1 Foo" << endl;}

virtual void Foo () {cout << "Def2 Foo" << endl;}

int main ()
{
  Def1 *pcDef1_1 = new Def1;
  Def1 *pcDef1_2 = new Def2;
  pcDef1_1->Foo();
  pcDef1_2->Foo();
  delete pcDef1_2;
  delete pcDef1_1;
}
```


Example

```
class Person
{
  private:
    string name;
  public:
    Person() { setName(""); }
    Person(string pName) { setName(pName); }
    void setName(string pName) { name = pName; }
    string getName() { return name; }
};
```

```
class Faculty : public Person
{
  private:
    Discipline department;
  public:
     Faculty(string fname, Discipline d)
       {setName(fname); setDepartment(d); }
     void setDepartment(Discipline d)
       { department = d; }
     Discipline getDepartment()
       { return department; }
};
```



```
class TFaculty : public Faculty
{
  private:
    string title;
  public:
    TFaculty(string fname, Discipline d, string title)
      : Faculty(fname, d)
    {
      setTitle(title);
    }
  void setTitle(string title) { this->title = title; }
   string getName() { return title + " " +
                      Person::getName(); }
};
```

Polymorphism in Action

- Is this code polymorphic? If not, how could we make it polymorphic?

```
const int NUM_PEOPLE = 5;

Person *arr[NUM_PEOPLE] = {
   new Tfaculty("Indiana Jones", ARCHEALOG, "Dr."),
   new Student("Thomas Cruise", COMPUTER_SCIENCE, NULL),
   new Faculty("James Stock", BIOLOGY),
   new Tfaculty("Sharon Rock", BIOLOGY, "Professor"),
   new TFaculty("Nicole Eweman", ARCHEOLOGY, "Dr,")};

for(int k = 0; k < NUM_PEOPLE; k++)

{

  cout << arr[k]->getName() << endl;

}
```