Intermediate/Advanced Computer Programming
Review

• How is the g++ compiler used to compile and link our program?

• What is the difference between compiling and linking?

• What is the difference between a class and an object?

• How is access controlled to object's member functions/data
  – BigInt Add(BigInt in);
Pointers and References
Passing Variables Into Functions

Pass by value

- Variables provided to the function are copies NOT the variables themselves

```c
int add(int a, int b)
{
    a = a + b
    return a;
}
```
Passing Variables Into/Out of Functions

```c
void add(int a, int b, int result)
{
    result = a + b
}
```
Pointers
void add(int a, int b, int *result)
{
    *result = a + b
}

Pointers
Pointers

Pointer

Data

(dereference)

* 

&

(address of)

Pointer

Data
Pointers

```cpp
int x = 5;
int *x_ptr = &x;

*x = 7;

cout << x << endl;
```
Pointers

• Why use pointers?
Const

Pointer

Data
const with Pointers

```c
int x = 5;

[const] int * [const] x_ptr = &x;
```
Pass by Reference

An **alias** for a variable
Acts as if the actual variable were used

```cpp
int x = 5;
int &x_ref = x;

x_ref = 15;

cout << x << endl;
```
void add(int a, int b, int &result)
{
    result = a + b
}

References
Function Overloading
Function Overloading

Consider the sqrt() functions

- double sqrt(double x);
- float sqrtf(float x);
- long double sqrtl(long double x);

Why not call all of these sqrt()?
Function Overloading

Using the same name for two or more functions

- Different parameter types
- Different number of parameters
- Cannot differ only in the return type
- Cannot differ only if one parameter is passed by-value and one parameter is passed by-reference

double sqrt(double x);
double sqrt(double x, int y);
float sqrt(float x);
long double sqrt(long double x);
Operator Overloading
BigInt a = "55";
BigInt b = "7";

BigInt c;

c = a + b;
Operator Overloading

Convert operator call into a function call and overload the function call

<BigInt> + <BigInt>
Operator Overloading

Convert operator call into a function call and overload the function call

\(<\text{BigInt}> + <\text{BigInt}>\)

\text{BigInt \ operator+}(\text{BigInt, BigInt});

- or -

\text{BigInt \ BigInt::operator+}(\text{BigInt});
BigInt a = "55";

BigInt c;

c = a + 7;