Physics and Computers

Lecturer: Jason Sang Hun Lee
Teaching assistant: Kijin Choi
Lecture times: 4pm-6pm, Mondays and Tuesdays

Teaching Team

- Lecturer: Jason Sang Hun Lee
  - Office: 14-323
  - Email: TBA
  - Phone: 6490-2656
- Teaching assistant: Kijin Choi
  - Office: 14-232
  - Email: unclok123@gmail.com
  - Phone: 6490-5751

Lecture outlines

1. Introduction to the lectures
2. Introduction to c++, g++ and hello world
3. Basic language features of c++
4. Data types and operators
5. Arrays, strings and vectors
6. Scope, loops and conditional statements
7. **Functions**
8. Pointers
9. Class
10. Input/Output
11. Encapsulation, inheritance and polymorphism
12. Templates

functions.c

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

int addition (int a, int b)
{
    int r;
    r=a+b;
    return r;
}
```

```
int main ()
{
    int z;
    z = addition (5,3);
    cout << "The result is " << z;
    return 0;
}
```

functions need to be declared before

functions.c

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

int global_x = 5;

int addition (int a, int b)
{
    int r;
    r=a+b+global_x;
    return r;
}

int main ()
{
    int a;
    a = addition (5,3) +global_x;
    cout << "The result is " << a;
    return 0;
}
```

be careful about scope

functions.c

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

int global_x = 5;

int addition (int a, int b)
{
    int r;
    r=a+b+global_x;
    return r;
}
void printmessage ()
{
    cout << "\nI'm a function!\n";
}
int main ()
{
    int a;
    a = addition (5,3) +global_x;
    cout << "The result is " << a;
    printmessage ();
    return 0;
}
```

void functions

functions.c

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

int global_x = 0;

int addition (int a, int b=10)
{
    int r;
    r=a+b+global_x;
    return r;
}
void printmessage ()
{
    cout << "\nI'm a function!\n";
}
int main ()
{
    int a;
    a = addition (5) +global_x;
    cout << "The result is " << a;
    printmessage ();
    return 0;
}
```

default values

functions.c

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

int operate (int a, int b)
{
    return (a*b);
}
float operate (float a, float b)
{
    return (a/b);
}
int main ()
{
    int x=5,y=2;
    float n=5.0,m=2.0;
    cout << operate (x,y);
    cout << "\n";
    cout << operate (n,m);
    cout << "\n";
    return 0;
}
```

overloaded functions

functions.c

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

long factorial (long a)
{
    if (a > 1)
        return (a * factorial (a-1));
    else
        return (1);
}

int main ()
{
    long number;
    cout << "Please type a number: ";
    cin >> number;
    cout << number << "! = " << factorial (number);
    return 0;
}
```

Recursivity

functions.c

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

long factorial (long a);

int main ()
{
    long number;
    cout << "Please type a number: ";
    cin >> number;
    cout << number << "! = " << factorial (number);
    return 0;
}
long factorial (long a)
{
    if (a > 1)
        return (a * factorial (a-1));
    else
        return (1);
}
```

declaring functions prototypes

quiz_03.c

```
Fibonacci series,
i = 5
the ith number is 5

Fibonacci series,
i = 20
the ith number is 6765
```

```
FFFFF
F
FFF
F
F

EEEEE
E
EEE
E
EEEEE
```