Workshop Boot Camp

☐ Write the function prototype of the void function named `hello()` that takes a string as a parameter.

```
void hello(string str);
```

☐ Write the function prototype of the double function named `area()` that takes two double as parameters.

```
double area(double x,double y);
```

☐ Write the function prototype of the string function named `concatenation()` that takes two string as parameters.

```
string concatenation(string x,string y);
```

☐ Write the function prototype of the double function named `average()` that takes five doubles as parameters.

```
double average(double x,double y,double z,double w,double v);
```

☐ Write the function prototype of the string function named `echo5()` that takes string as a parameter.

```
string echo5(string x);
```

☐ Write the function prototype of the char function named `grade()` that takes double as a parameter.

```
char grade(double x);
```

☐ Write the function prototype of the void function named `swap()` that takes two int references as parameters.

```
void swap(int& x,int& y);
```

☐ Write the function prototype of the int function named `median()` that takes three ints as parameters.

```
int median(int x,int y,int z);
```

☐ Write the function prototype of the void function named `power()` that takes a constant int reference, int reference and an int as parameters.

```
void power(const int& x,int& y,int z);
```

☐ Write the function definition of `hello()` that displays the message "Hello, " followed by the string parameter.

```
void hello(string x)
{
 cout << "Hello, " << x;
}
```

☐ Write the function definition of `area()` that returns the product of the parameters.

```
double area(double x,double y)
{
 return x * y;
}
```

☐ Write the function definition of `concatenation()` that returns the concatenation of the first parameter to the end of the second parameter.

```
string concatenation(string x,string y)
{
 return (y + x);
}
```

☐ Write the function definition of `average()` that returns the average of the parameters.

```
double average(double x,double y,double z,double w,double v)
{
 return (x + y + z + w + z) / 5.0;
}
```