Function

Two types of functions:

- **void function with value and reference parameters**
- **value returning function**

Function Parameter

- **Parameter passing by value**
  - a copy of the data is created and placed in a local variable in the called function
  - regardless how the data is manipulated and changed in the called function, the original data in the called function are safe and unchanged
- **Parameter passing by reference : necessary when more than one value need to be passed back to the calling function**
  - sends the address of a variable to the called function, rather than sending its value
  - used when you want to change the content of a variable in the calling function
    Indicate reference parameter(s) by adding the address operator : **&**

Example 1

```
void ComputePrice(char, int);
int main( )
{
      char   packageChoice;
      int     age;

      cout << "which package do you like " << endl;
      cin >> packageChoice;

      cout << "how old are you?" << endl;
      cin >> age;

      ComputePrice (packageChoice, age);

      return 0;
}

void ComputePrice(char  choice,   int  age)
{
       float  price;

      if (choice == 's')
          price = 500.0;
      else if (choice == 'p')
          price = 620.0;
      else if (choice == 'v')
          price = 850.0;

       if  (age <= 2)
           price = 0.0;
       else if (age <= 18)
           price = price/2.0;
       else if (age >= 65)
           price = price * 0.9;

       cout << "your price is " << price << endl;
       return ;
}
```

Example 2:

```
void Exchange (int &, int &);
int main()
{
        int  num1= 3,  num2 = 5;

        cout << num1 << "\t" << num2 << endl;

        Exchange (num1, num2);

        cout << num1 << "\t" << num2  << endl;

        return 0;
}

void Exchange (int &  number1,  int &  number2)
{
        int  temp;

        temp = number1;
        number1 = number2;
        number2 = temp;

        return;
}
```

Example 3:

```
void Divide(int, int, int&, int&);
int main( )
{
        int  num1, num2;
        int  quotient, remainder;

        cout << "Enter two integers\n" ;
        cin >> num1 >> num2;

        Divide(num1, num2, quotient, remainder);

        cout << "Quotient is " << quotient << "\t";
        cout << "Remainder is " << remainder
             << endl;

        return 0;
}

void Divide(int  numerator,  int  denominator,  int  quotient,   int  & remainder)
{
        quotient  = numerator / denominator;
        remainder = numerator % denominator;

        return;
}
```

Practice question1: write a user defined function DrawRectangle

```
#include <iostream>
using namespace std;
// write function declaration here

_____
int main()
{
        int length, width;

        cout << "please enter the length and width of the rectangle";
        cin >> length >> width;

        // write function call here to draw the rectangle on screen

       _____

        return 0;
}

// write function definition here
```

Practice question2: write user defined function **GetInformation** and **ComputeBalance**

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

// write function declarations here

_____

_____

int main()
{
        float   balance, deposit, withdraw;

        // call function  GetInformation here to
        // input these information from user: initial balance, amount of deposit, amount of withdraw

       _____


        // call function ComputeBalance here to
        // compute the current balance of the account

        _____


        // display information
         cout<< "This month, total deposit is " << deposit << endl;
         cout << "              total withdraw is " << withdraw << endl;
         cout << "              balance at the end of the month is " << balance << endl;

         return 0;
}

// write function definitions for GetInformation and ComputeBalance below
```

Value returning function

- **A value is explicitly returned using "return" statement**
  The value can be of any C++ data type: char, int, float, bool, string
  The return type in the function header and function declaration should correspond to the type of the value returned
- **A value returning function is activated/called within an expression.**
  (the value returned will be used in evaluating the expression)
  (Often, the function activation is the expression by itself)

```
// function declaration


int main()
{
    float   tempIn, tempOut;
    char type;

    cout << "please enter the temperature in the form: 34 F,  or 59 C : ";
    cin   >> tempIn >> type;

    // activate the function to convert the temperature

    _____________________________________

    if (type == 'F')
        cout << tempIn << " Fahrenheit equals to " << tempOut << " Celsius" << endl;
    else
        cout << tempIn << " Celsius equals to " << tempOut << " Celsius" << endl;

    return 0;
}

// function definition
float  Convert(float temp, char type)
{
    float convertedTemp;

    if (type == 'F')
        convertedTemp = 9 * temp / 5 + 32;
    else
        convertedTemp = 5 * (temp – 32) / 9;

     return convertedTemp;
}
```

Practice Questions: Write a C++ value returning function that

1. takes the length of the two sides of a right triangle, and computes and returns the perimeter of the triangle
2. receives a floating-point number and returns the fractional part of that number. For example, if the incoming value of x is 16.753, the function returns the value 0.753.
3. returns the smallest of three integer parameters.
4. determines whether a character entered is an alpha numeric character. Returns true if it is an alpha numeric character, returns false otherwise.
5. determines whether an integer value is a prime number. Returns true if it is a prime number, returns false if it is not.

Can I use reference parameter with value returning function?

When to use value-returning function, and when to use void function?