

## Essentials of a Function

1. Prototype
2. Call (Invocation)
3. Definition

```
bool Is_A_Grade (int Scr);
int main()
{   int Score = 85;
    if (Is_A_Grade(Score))
        cout << "Grade is A" << endl;
    return EXIT_SUCCESS;
}
bool Is_A_Grade(int Scr)
{   if (Scr >= 90)
        return true;
    else return false;
}
```

October 6

CS 1344

## Functions - Cont.

Prototype

`bool Is_A_Grade (int Scr);`

Return type

Name

ParameterType ParameterName (optional) ....  
(Formal Parameter)

Call (Invocation)

`Is_A_Grade (Score);`

Function Name

Actual Parameter

October 6

CS 1344

## Functions - Cont.

Definition

ParameterType ParameterName (required) ....  
(Formal Parameter(s))

Return type

Name

`bool Is_A_Grade(int Scr)`

```
{
    if (Scr >= 90)
        return true;
    else return false;
}
```

Return Value

October 6

CS 1344

## Parameter Passing - By Value

```
void OutputData(int A, int B, int C);
int main()
{
    int A1, B1, C1;
    cin >> A1 >> B1 >> C1;
    OutputData(A1, B1, C1);
}
void OutputData(int A, int B, int C)
{
    cout << A << B << C << endl;
}
```

October 6

CS 1344

# Parameter Passing - By Reference

---

```
void ReadData(int& A, int& B, int& C);  
int main()  
{  
    int A1, B1, C1;  
    ReadData(A1, B1, C1);  
    OutputData(A1, B1, C1);  
}  
void ReadData(int& A, int& B, int& C)  
{  
    cin >> A >> B >> C;  
}
```