```
//1: Pass-by-Value (int)
#include <iostream>
using namespace std;

// Function to calculate the square of an integer (pass-by-value)
int square(int num) {  // Function declaration: 'num' is passed by value, so a copy is made
   return num * num;  // Calculates the square of 'num' and returns the result
}

int main() {
   int value = 5;  // Declare an integer variable 'value' and initialize it to 5
   int result = square(value);  // Passing 'value' to the 'square' function by value

   cout << "The square of " << value << " is: " << result << endl;  // (it is an object
belongs to ostream class "<<" refers to bits are sending from memory to desktop)
   return 0;  // Indicates successful program termination
}
```

==========

```
//2 Pass-by-Reference (int)
#include <iostream>
using namespace std;

// Function to double the value using pass-by-reference
void doubleValue(int &num) {  // 'num' is passed by reference, allowing direct modification
of the original variable
   num = num * 2;  // The original 'num' value is doubled, modifying the value in memory
directly
}

int main() {
   int value = 5;  // Declare and initialize the variable 'value' with 5
   doubleValue(value);  // Passing 'value' by reference to the function, where it will be
modified

   cout << "The doubled value is: " << value << endl;  // (it is an object belongs to
ostream class "<<" refers to bits are sending from memory to desktop)
   return 0;  // Exit the program
}
```

==========

```
// 3 Pass-by-Value (std::string)
#include <iostream>  // Includes the header for input/output operations
#include <string>    // Includes the header for string operations
using namespace std;  // Makes standard library names accessible without std:: prefix

// Function to concatenate a string with " World" (pass-by-value)
string concatenate(const string str) {  // 'str' is passed by value (a copy of the string is
made)
```

```
    return str + " World";  // Concatenates the string with " World" and returns the result
}

int main() {
    string greeting = "Hello";  // Declares and initializes the string variable 'greeting'
    string newGreeting = concatenate(greeting);  // Passing 'greeting' by value to the
function 'concatenate'

    cout << newGreeting << endl;  // (it is an object belongs to ostream class "<<" refers to
bits are sending from memory to desktop)
    return 0;  // Returns 0 to indicate that the program ended successfully
}
```

==========

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

// Function to append a string with " World" (pass-by-reference)
void appendWorld(string &str) {  // 'str' is passed by reference
    str += " World";  // Modifies the actual string 'str', appending " World"
}

int main() {
    string greeting = "Hello";  // Declare a string variable 'greeting' initialized to "Hello"
    appendWorld(greeting);  // Passing 'greeting' by reference to append " World"

    cout << greeting << endl;  // (it is an object belongs to ostream class "<<" refers to bits
are sending from memory to desktop)
    return 0;  // Return 0 to indicate the program has ended successfully
}
```

==========

```
//5 Pass-by-Value (float)
#include <iostream>
using namespace std;

// Function to calculate the area of a circle (pass-by-value)
float calculateArea(float radius) {  // 'radius' is passed by value
    return 3.14 * radius * radius;  // Calculates area of a circle
}
int main() {
    float r = 4.0;  // Declaring a variable 'r' with value 4.0 for radius
    float area = calculateArea(r);  // Passing 'r' by value to the function 'calculateArea'

    cout << "The area of the circle is: " << area << endl;  // (it is an object belongs to
ostream class "<<" refers to bits are sending from memory to desktop)
    return 0;  // Exiting the main function
}
```