Pass by reference v. Pass by value For the ENGR 101 GSI Application

Johnson He (hejohns)

March 19, 2023



Pass by Value

Pass by Value

```
void say_my_name(string name) {
   name = "I'm " + name + "!";
   cout << name << endl;
}</pre>
```

stdout:

```
void say_my_name(string name) {
    name = "I'm " + name + "!";
    cout << name << endl;
int main(){
    string me = "Johnson";
    say_my_name (me);
    return 0;
```

Johnson He (hejohns)

stdout:

```
void say_my_name(string name) {
    name = "I'm " + name + "!";
    cout << name << endl;
int main(){
\rightarrow
    string me = "Johnson";
    say_my_name (me);
    return 0;
```

Johnson He (hejohns)

```
void say_my_name(string name) {
    name = "I'm " + name + "!";
    cout << name << endl;
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name (me);
    return 0;
stdout:
```

Johnson He (hejohns)

stdout:

```
void say_my_name(string name) {
    name = "I'm " + name + "!";
    cout << name << endl;
int main(){
    string me = "Johnson";
    // me = "Johnson"
  say_my_name(me);
    return 0;
```

Johnson He (hejohns) GSI Application March 19, 2023

2/4

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    cout << name << endl;
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name (me);
    return 0;
stdout:
```

Johnson He (hejohns)

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
→ // name = "I'm Johnson!"
    cout << name << endl;
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
    return 0;
stdout:
```

Johnson He (hejohns)

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl;
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
    return 0;
```

stdout: I'm Johnson!

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
   // me = "Johnson"
    return 0;
```

stdout: I'm Johnson!

Α

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
    // me = "Johnson"
    return 0:
```

stdout: I'm Johnson!

Pass by Value

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name (me);
    // me = "Johnson"
    return 0:
```

Pass by Reference

Α

```
void say_my_name(string& name) {
   name = "I'm " + name + "!";
   cout << name << endl;
}</pre>
```

stdout: I'm Johnson!

```
Pass by Value
```

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name (me);
    // me = "Johnson"
    return 0:
```

```
void say_my_name(string& name) {
          name = "I'm " + name + "!";
          cout << name << endl;
      int main(){
          string me = "Johnson";
          say_my_name(me);
          return 0;
```

stdout: I'm Johnson!

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name (me);
```

```
void say_my_name(string& name) {
          name = "I'm " + name + "!";
           cout << name << endl;
      int main(){
      \rightarrow
           string me = "Johnson";
           say_my_name(me);
           return 0;
```

stdout: I'm Johnson!

return 0:

// me = "Johnson"

```
Pass by Value
```

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name (me);
    // me = "Johnson"
    return 0:
```

```
void say_my_name(string& name) {
          name = "I'm " + name + "!";
          cout << name << endl;
      int main(){
          string me = "Johnson";
          // me = "Johnson"
          say_my_name(me);
          return 0;
```

stdout: I'm Johnson!

GSI Application

```
Pass by Value
```

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name (me);
    // me = "Johnson"
    return 0:
```

```
void say_my_name(string& name) {
          name = "I'm " + name + "!";
          cout << name << endl;
      int main(){
          string me = "Johnson";
          // me = "Johnson"
          say_my_name(me);
          return 0;
```

stdout: I'm Johnson!

Johnson He (hejohns) GSI App

March 19, 2023

2/4

```
void say_my_name(string& name) {
   // name = "Johnson"
    name = "I'm " + name + "!";
    cout << name << endl;
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
    return 0;
```

stdout: I'm Johnson!

```
void say_my_name(string& name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
   // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
    return 0;
```

stdout: I'm Johnson!

```
void say_my_name(string& name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
    return 0;
```

stdout: I'm Johnson!

stdout: I'm Johnson!

```
void say_my_name(string name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name (me);
    // me = "Johnson"
    return 0:
```

```
void say_my_name(string& name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!"
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
    // me = "I'm Johnson!"
    return 0;
```

stdout: I'm Johnson!

2/4

stdout: I'm Johnson!

```
stdout: I'm Johnson!
```

return 0:

Pass by Reference

```
void say_my_name(string& name) {
    // name = "Johnson"
    name = "I'm " + name + "!";
    // name = "I'm Johnson!" = me
    cout << name << endl:
int main(){
    string me = "Johnson";
    // me = "Johnson"
    say_my_name(me);
    // me = "I'm Johnson!"
    return 0;
```

stdout: I'm Johnson!

say_my_name (me);

// me = "Johnson"

Example 1

By Value

3/4

By Value

By Reference

Α

By Value

By Reference

Α

By Const Reference

```
bool
same_first(
   const string& lhs,
   const string& rhs){
   return !lhs.empty() &&
       !rhs.empty() &&
       lhs.at(0) ==
       rhs.at(0);
```

```
By Value
```

By Reference

By Const Reference

```
bool
same_first(string& lhs,
string& rhs){
return !lhs.empty() &&
!rhs.empty() &&
lhs.at(0) ==
rhs.at(0);
}
```

Α

```
bool
same_first(
    const string& lhs,
    const string& rhs) {
    return !lhs.empty() &&
        !rhs.empty() &&
        lhs.at(0) ==
        rhs.at(0);
}
```

Example 2

By Value

```
void
redact(string msg) {
  const string secret =
    "SECRET";
  auto pos =
    msg.find(secret);
  if(pos != npos) {
    msq.replace(n,
      secret.size(),
      "REDACTED");
```

By Value

By Reference

```
void
redact(string msg) {
  const string secret =
    "SECRET";
  auto pos =
    msq.find(secret);
  if (pos != npos) {
    msg.replace(n,
      secret.size(),
      "REDACTED");
```

```
void
redact(string& msg) {
  const string secret =
    "SECRET";
  auto pos =
    msq.find(secret);
  if (pos != npos) {
    msg.replace(n,
      secret.size(),
      "REDACTED");
```

Α

```
void
redact(string msg) {
  const string secret =
    "SECRET";
  auto pos =
    msq.find(secret);
  if (pos != npos) {
    msg.replace(n,
      secret.size(),
      "REDACTED");
```

```
void
redact(string& msg) {
  const string secret =
    "SECRET";
  auto pos =
    msq.find(secret);
  if (pos != npos) {
    msg.replace(n,
      secret.size(),
      "REDACTED");
```

Α

```
void
redact (
  const string& msg) {
  const string secret =
    "SECRET":
  auto pos =
    msq.find(secret);
  if(pos != npos) {
    msg.replace(n,
      secret.size(),
      "REDACTED");
```

```
void
redact(string msg) {
  const string secret =
    "SECRET";
  auto pos =
    msq.find(secret);
  if (pos != npos) {
    msg.replace(n,
      secret.size(),
      "REDACTED");
```

```
void
redact(string& msg) {
  const string secret =
    "SECRET";
  auto pos =
    msq.find(secret);
  if (pos != npos) {
    msg.replace(n,
      secret.size(),
      "REDACTED");
```

```
void
redact (
  const string& msg) {
  const string secret =
    "SECRET":
  auto pos =
    msq.find(secret);
  if(pos != npos) {
    msq.replace(n,
      secret.size(),
      "REDACTED");
```