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;

}
```

Pass by Value

```
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:

Pass by Value

```
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:

Pass by Value

```
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:

Pass by Value

```
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:

Pass by Value

```
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:

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);

    return 0;
}
```

---

stdout:

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);

    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;
}
```

---

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;
}
```

stdout: I'm Johnson!

Pass by Reference

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
void say_my_name(string& name){

    name = "I'm " + name + "!";

    cout << name << endl;

}
```

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
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:

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
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:

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
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:

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
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:

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
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:

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
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:

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
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!

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;
}
```

stdout: I'm Johnson!

Pass by Reference

```
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!

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;
}
```

stdout: I'm Johnson!

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!

Example 1

By Value

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

Example 1

By Value

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

By Reference

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

Example 1

By Value

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

By Reference

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

By Const Reference

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

Example 1

By Value

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

By Reference

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

By Const Reference

```
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){
    msg.replace(n,
      secret.size(),
      "REDACTED");
  }
}
```

Example 2

By Value

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

By Reference

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

Example 2

By Value

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

By Reference

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

By Const Reference

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

Example 2

By Value

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

By Reference

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

By Const Reference

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