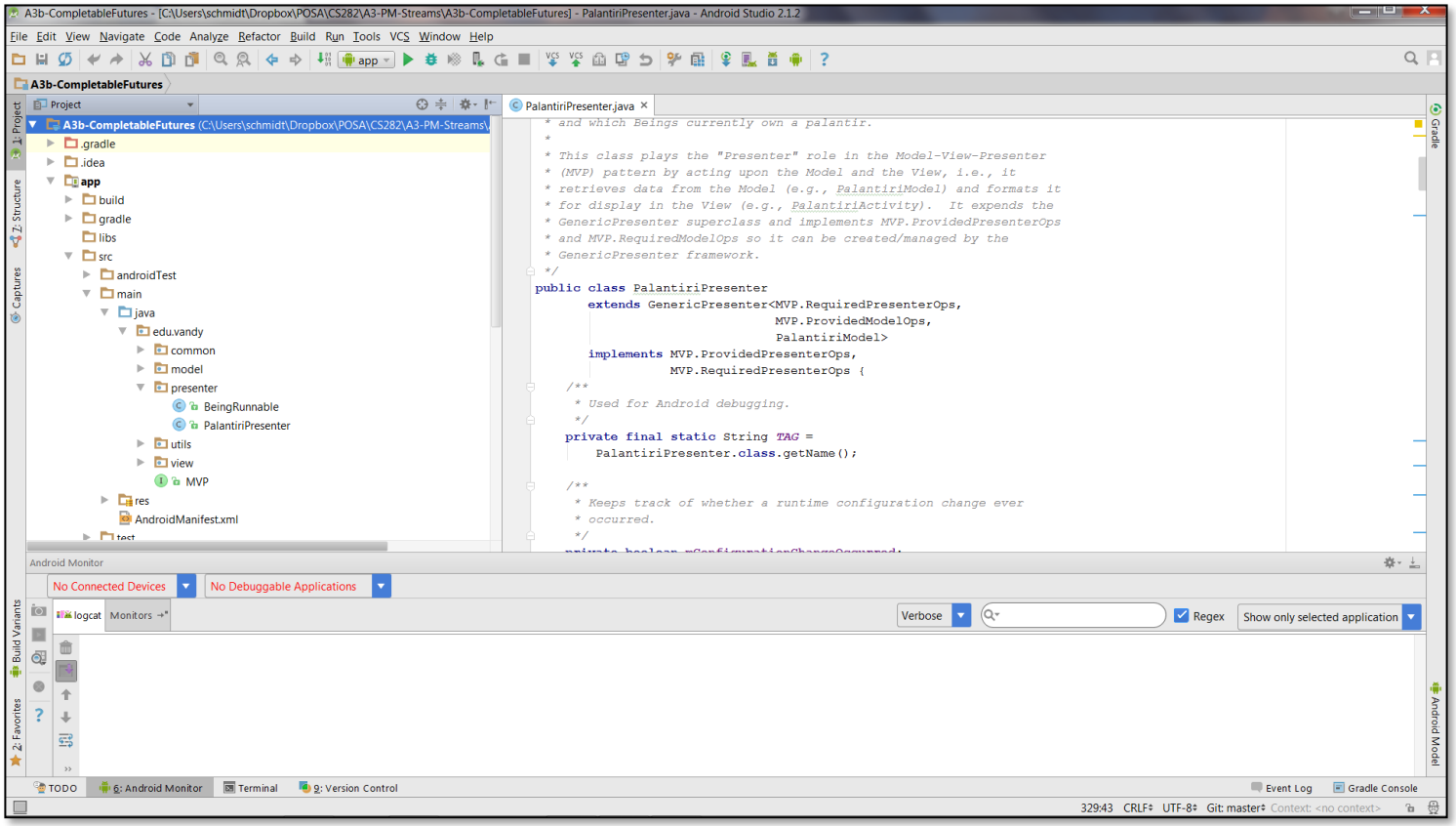
The Template Method Pattern

Implementation in C++

Douglas C. Schmidt

Learning Objectives in This Lesson

- Recognize how the *Template Method* pattern can be applied to flexibly support multiple operating modes in the expression tree processing app.
- Understand the structure & functionality of the *Template Method* pattern.
- Know how to implement the *Template Method* pattern in C++.



Template Method example in C++

- Allow subclasses to customize certain steps in the input handling algorithm.

```
class ET_Event_Handler : public Event_Handler {
  ...
  void handle_input() override {
    prompt_user();
    string input = receive_input();
    User_Command command = make_command(input);
    execute_command(command);
  }
```

Template method

Template Method example in C++

- Allow subclasses to customize certain steps in the input handling algorithm.

```
class ET_Event_Handler : public Event_Handler {
  ...
  void handle_input() override {
    prompt_user();
    string input = receive_input();
    User_Command command = make_command(input);
    execute_command(command);
  }
```

Hook methods

Template Method example in C++

- Allow subclasses to customize certain steps in the input handling algorithm.

```
class ET_Event_Handler : public Event_Handler {
  ...
  void handle_input() override {
    prompt_user();
    string input = receive_input();
    User_Command command = make_command(input);
    execute_command(command);
  }

  unique_ptr<Event_Handler> make_handler(bool verbose) {
    return verbose
      ? make_unique<Verbose_Mode_ET_Event_Handler>()
      : make_unique<Succint_Mode_ET_Event_Handler>();
  }
```

Factory method creates designated concrete classes

This is not the only/best way to define a factory since it's too tightly coupled.

Template Method example in C++

- Allow subclasses to customize certain steps in the input handling algorithm.

```
class Verbose_Mode_ET_Event_Handler : public ET_Event_Handler {
  ...
  public User_Command make_command(string user_input) {
    return command_factory_.make_command
                (user_input);
  }
```

Specialized hook method

Template Method example in C++

- Allow subclasses to customize certain steps in the input handling algorithm.

```
class Verbose_Mode_ET_Event_Handler : public ET_Event_Handler {
  ...
  public User_Command make_command(string user_input) {
    return command_factory_.make_command
                (user_input);
  }
```

```
class Succinct_Mode_ET_Event_Handler : public ET_Event_Handler {
  ...
  public User_Command make_command(string user_input) {
    return command_factory_.make_macro_command
                (user_input);
  }
```

Specialized hook method

Template Method example in C++

- Allow subclasses to customize certain steps in the input handling algorithm.

```
class Verbose_Mode_ET_Event_Handler : public ET_Event_Handler {
  ...
  public User_Command make_command(string user_input) {
    return command_factory_.make_command
                (user_input);
  }


class Succinct_Mode_ET_Event_Handler : public ET_Event_Handler {
  ...
  public User_Command make_command(string user_input) {
    return command_factory_.make_macrp+command
                (user_input);
  }
```

See earlier lesson on "*The Factory Method Pattern*" for `User_Command_Factory`