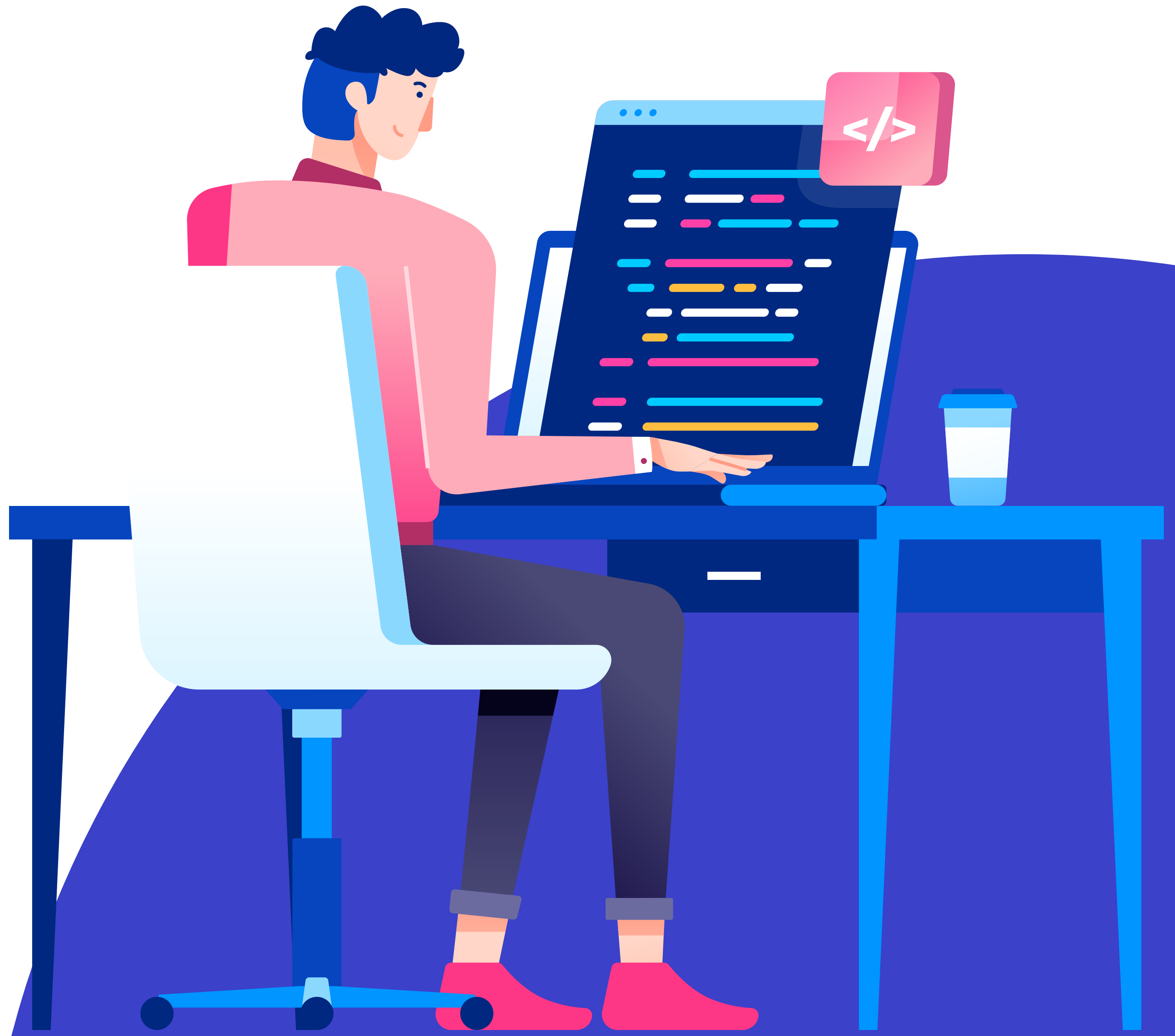
Search . . .

ARMADA

Finals Presentation

Prepared by:
Daskeo, Kristine M.
Manalo, John Christian A.
Ramos, Julianne Adrielle T.



Search . . .

ARMADA

```
require 'spec_helper'
require 'rspec/rails'

require 'capybara/rspec'
require 'capybara/rails'

Capybara.javascript_driver =
Category.delete_all; Category
Shoulda::Matchers.configure do |config|
  config.integrate do |with|
    with.test_framework :rspec
    with.library :rails
  end
end

# Add additional requires below this line.
# Requires supporting ruby files
# spec/support/ and its subdirectories
# run as spec files by default.
# in _spec.rb will both be required
# run twice. It is recommended
# end with _spec.rb. You can configure
```

It aims to facilitate the **development of reliable and high-integrity systems**, as well as **resilient to failures and meets stringent requirements for critical applications**.

This makes it an ideal choice for industries **where reliability and security are paramount**, such as aerospace, automotive, and healthcare, **where ensuring the correctness and stability of systems is crucial**.

Search . . .

ARMADA CONSTRUCTS

DATA TYPES

coords
status
string
double

EXPRESSION

Mach

FUNCTION

print

CONDITIONAL

case

OBJECT

create

Search . . .

ARMADA SYNTAX CHECKER

CFG DEFINITION

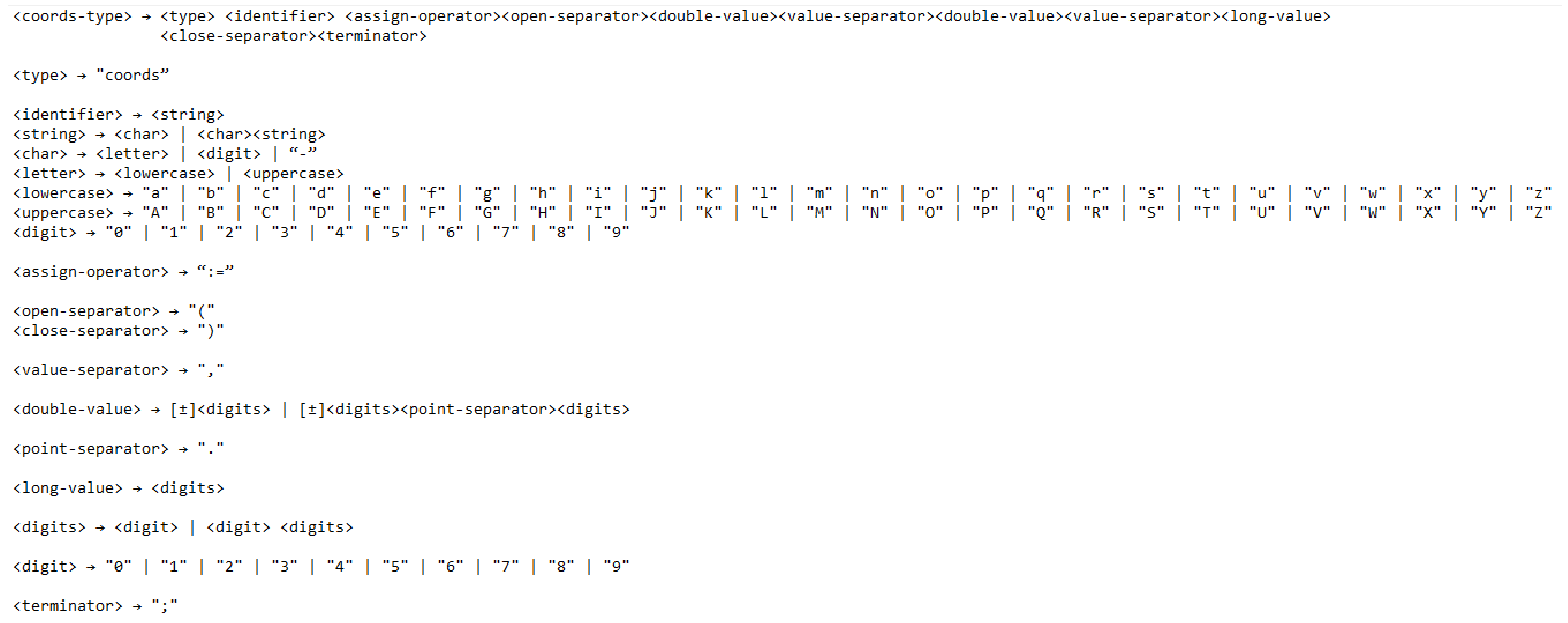
```
<MainCFG> → <coords-type> | <mach-expression> | <print-function>

<IfCFG> → <case-conditional>

<ObjectCFG> → <create-declaration>
```

CCPGLANG COM211

Search . . .

DATA TYPES COORDINATE

CFG DEFINITION



```
<coords-type> → <type> <identifier> <assign-operator><open-separator><double-value><value-separator><double-value><value-separator><long-value>
                <close-separator><terminator>

<type> → "coords”

<identifier> → <string>
<string> → <char> | <char><string>
<char> → <letter> | <digit> | “-”
<letter> → <lowercase> | <uppercase>
<lowercase> → "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
<uppercase> → "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
<digit> → "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<assign-operator> → “:=”

<open-separator> → "("
<close-separator> → ")"

<value-separator> → ","

<double-value> → [±]<digits> | [±]<digits><point-separator><digits>

<point-separator> → "."

<long-value> → <digits>

<digits> → <digit> | <digit> <digits>

<digit> → "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<terminator> → ";"
```

Search . . .

DATA TYPES COORDINATE

TEST CASES

Valid Input:

```
coords Location := (14.5123, 121.0665, 13);
END
```

Expected Output:

```
Location of type coordinates is set to (latitude = 14.5123, longitude = 121.0665, altitude = 13)
```

Search . . .

DATA TYPES COORDINATE

TEST CASES

Invalid Input:

```
Coords Location := (14.5123, 121.0665, 13);
END
```

Expected Output:

```
Error: Invalid syntax -> Coords Location := (14.5123, 121.0665, 13);
Check for missing or misplaced semi-colons and correct variable names.
```

Search . . .

DATA TYPES COORDINATE

TEST CASES

Invalid Input:

```
coords loc := (131.41221, 413123.43, 131.31);
END
```

Expected Output:

```
Error: Wrong number format. It must be (double latitude, double longitude, long altitude).
```

Search . . .

DATA TYPES COORDINATE

TEST CASES

Invalid Input:

```
coords loc := (14.000, 120.000);
END
```

Expected Output:

```
Error: Missing value. It must be (double latitude, double longitude, long altitude).
```

CCPGLANG COM211

Search . . .

EXPRESSION MACH

CFG DEFINITION

```
<mach-expression> → <data-type> <identifier> <assign-operator> <keyword><open-separator><double-value><value-separator><double-value><close-separator><terminator>

<data-type> → “double”

<identifier> → <string>
<string> → <char> | <char><string>
<char> → <letter> | <digit> | “-”
<letter> → <lowercase> | <uppercase>
<lowercase> → "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
<uppercase> → "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
<digit> → "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<assign-operator> → “:=”

<keyword> → “Mach”

<open-separator> → “(“
<close-separator> → “)”

<value-separator> → “,”

<double-value> → [±]<digits> | [±]<digits><point-separator><digits>
<digits> → <digit> | <digit> <digit>
<digit> → "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<terminator> → ";"
```

Search . . .

EXPRESSION MACH

TEST CASES

Valid Input:

```
double Speed := Mach(25.0, 5.0);
END
```

Expected Output:

```
Speed is equal to 5.0
```

Search . . .

EXPRESSION MACH

TEST CASES

Invalid Input:

```
double speed := Mach(21);
END
```

Expected Output:

```
Error: Missing values for Mach. It must be (double value1, double value2).
```

Search . . .

EXPRESSION MACH

TEST CASES

Invalid Input:

```
long Speed := Mach(25.0, 5.0);
END
```

Expected Output:

```
Error: Invalid syntax -> long Speed := Mach(25.0, 5.0);
Check for missing or misplaced semi-colons and correct variable names.
```

CCPGLANG COM211

Search . . .

FUNCTION PRINT

CFG DEFINITION

```
<print-function> → <keyword><open-separator><print-stmt><close-separator><terminator>

<print-stmt> → <print-string> | <print-identifier>

<print-string> → <quote-separator><string><quote-separator>

<print-identifier> → <string>

<keyword> → “print”

<open-separator> → “(“
<close-separator> → “)”

<string> → <char> | <char><string>
<char> → <letter> | <digit> | “-”
<letter> → <lowercase> | <uppercase>
<lowercase> → "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
<uppercase> → "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
<digit> → "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<terminator> → ";"
```

Search . . .

FUNCTION PRINT

TEST CASES

Valid Input:

```
print("Hello World!");
END
```

Expected Output:

```
Hello World!
```

Search . . .

FUNCTION PRINT

TEST CASES

Valid Input:

```
coords Location := (14.5123, 121.0665, 13);
print(Location);
END
```

Expected Output:

```
Location of type coordinates is set to (latitude = 14.5123, longitude = 121.0665, altitude = 13)
(latitude = 14.5123, longitude = 121.0665, altitude = 13)
```

Search . . .

FUNCTION PRINT

TEST CASES

Invalid Input:

```
Print("Hello World!");
END
```

Expected Output:

```
Error: Invalid syntax -> Print("Hello World!");
Check for missing or misplaced semi-colons and correct variable names.
```

CCPGLANG COM211

Search . . .

FUNCTION PRINT

TEST CASES

Invalid Input:

```
print(location);
END
```

Expected Output:

```
Error: Identifier 'location' not found. Check variable declarations.
```

Search . . .

FUNCTION PRINT

TEST CASES

Invalid Input:

```
print (“Hello World!”);
END
```

Expected Output:

```
Error: Invalid syntax -> print (“Hello World!”);
Check for missing or misplaced semi-colons and correct variable names.
```

CCPGLANG COM211

Search . . .

CONDITIONAL CASE

CFG DEFINITION

```
<case-conditional> → <keyword> <open-separator><conditions>
<close-separator><open-brace><statements>
<close-brace>

        <keyword> → “create”

<open-separator> → “(“
<close-separator> → “)”

<conditions> → <simple-condition> | <compound-condition>
<simple-condition> → <identifier> <operator> <identifier>

<compound-condition> → <simple-condition> ( <logical-operator>
<simple-condition> )*

<operator> → "==" | "!=" | ">" | "<" | ">=" | "<="

<logical-operator> → "&&" | "||"

<identifier> → <string>
<string> → <char> | <char><string>
<char> → <letter> | <digit> | “-”
<letter> → <lowercase> | <uppercase>
<lowercase> → "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
<uppercase> → "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
<digit> → "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<terminator> → “;”
```

Search . . .

CONDITIONAL CASE

TEST CASES

Valid Input:

```
case (a == b) {
 // statements
}
case (x > 10) {
 // statements
}
case (value >= 20) {
 // statements
}
END
```

Expected Output:

```
Valid case statement: case (a == b) {
Valid case statement: case (x > 10) {
Valid case statement: case (value >= 20) {
```

Search . . .

CONDITIONAL CASE

TEST CASES

Valid Input:

```
status flightStatus;
coords location := (0, 0, 0);
coords NAIA := (14.5123, 121.0165, 13);
coords DIA := (7.122552, 125.64550, 22);

case (location == NAIA) {
 flightStatus := "Landed";
}
case (location != NAIA && location != DIA) {
 flightStatus := "Airborne";
}
case (location == DIA) {
 flightStatus := "Boarding";
}
END
```

Expected Output:

```
Data type declared flightStatus of type status.
location of type coordinates is set to (latitude
= 0.0, longitude = 0.0, altitude = 0)
NAIA of type coordinates is set to (latitude =
14.5123, longitude = 121.0165, altitude = 13)
DIA of type coordinates is set to (latitude =
7.122552, longitude = 125.6455, altitude = 22)

Valid case statement: case (location == NAIA) {
flightStatus of type status is set to Landed.
Valid case statement: case (location != NAIA &&
location != DIA) {
flightStatus of type status is set to Airborne.
Valid case statement: case (location == DIA) {
flightStatus of type status is set to Boarding.
```

Search . . .

CONDITIONAL CASE

TEST CASES

Invalid Input:

```
case (x != ) {
 // stmts
}
case (temp <) {
 // stmts
}
END
```

Expected Output:

```
Invalid case statement (invalid condition): case (x != ) {
Invalid case statement (invalid condition): case (temp <) {
```

Search . . .

CONDITIONAL CASE

TEST CASES

Invalid Input:

```
case (a == b) {
END
```

Expected Output:

```
Invalid case statement (missing closing brace): case (a == b) {
```

Search . . .

CONDITIONAL CASE

TEST CASES

Invalid Input:

```
case (location) {
}
END
```

Expected Output:

```
Invalid case statement (invalid condition): case (location) {
```

Search . . .

OBJECT CREATE

CFG DEFINITION

```
<create-declaration> → <keyword> <identifier><open-brace><fields><close-brace>

<keyword> → “create”

<open-brace> → “{“
<close-brace> → “}“
<fields> → <field> | <field><fields>
<field> → <data-type> <identifier><terminator>

<data-type> → “string” | “coords” | “status”

<identifier> → <string>
<string> → <char> | <char><string>
<char> → <letter> | <digit> | “-”
<letter> → <lowercase> | <uppercase>
<lowercase> → "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"
<uppercase> → "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"
<digit> → "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<terminator> → “;”
```

Search . . .

OBJECT COORDINATE

TEST CASES

Valid Input:

```
create Object02{
   double speed;
   status flight_status;
}
END
```

Expected Output:

```
create Object02{
 double speed;
 status flight_status;
}
END
Valid object creation: Object02{
Field added: speed of type double.
Field added: flight_status of type status.
Object creation completed.
```

Search . . .

OBJECT COORDINATE

TEST CASES

Valid Input:

```
create Object {
 string name;
 coords location;
 double speed;
 status flightStatus;
}
Object airplane;
airplane.name := "Boeing";
airplane.location := (14.5123, 121.0165, 13);
airplane.speed := Mach(25.0, 5.0);
airplane.flightStatus := "Landed";
print(airplane.name);
print(airplane.location);
print(airplane.speed);
END
```

Expected Output:

```
Valid object creation: Object
Field added: name of type string.
Field added: location of type coords.
Field added: speed of type double.
Field added: flightStatus of type status.
Object creation completed.
Object instance created: airplane of type Object
Assigned string to airplane.name
Assigned coords to airplane.location
Assigned Mach result to airplane.speed
Assigned status to airplane.flightStatus
Field value for airplane.name: Boeing
Field value for airplane.location: Coordinates
(latitude=14.5123, longitude=121.0165, altitude=13)
Field value for airplane.speed: 125.0
```

Search . . .

OBJECT COORDINATE

TEST CASES

Invalid Input:

```
create Object {
 string name;
 coords location;
 double speed;
 status flightStatus;
END
```

Expected Output:

```
Valid object creation: Object
Field added: name of type string.
Field added: location of type coords.
Field added: speed of type double.
Field added: flightStatus of type status.
Error: Object creation incomplete, missing closing
brace.
```

Search . . .

OBJECT COORDINATE

TEST CASES

Invalid Input:

```
airplane.name := "Boeing";
airplane.location := (14.5123, 121.0165, 13);
airplane.speed := Mach(25.0, 5.0);
airplane.flightStatus := "Landed";
END
```

Expected Output:

```
Error: Object airplane not found.
Error: Object airplane not found.
Error: Object airplane not found.
Error: Object airplane not found.
```

Search . . .

OBJECT COORDINATE

TEST CASES

Invalid Input:

```
create Object {
 status flightStatus;
}
Object airplane;
airplane.flightStatus := "Delayed";
END
```

Expected Output:

```
Valid object creation: Object
Field added: flightStatus of type status.
Object creation completed.
Object instance created: airplane of type Object
Error: Invalid status -> "Delayed". Valid statuses
are 'Landed', 'Airborne', and 'Boarding'.
```

CODING School

Search . . .

THANK YOU