SAP CODE

BASIC OOPS

```
CLASS MCA DEFINITION.
PUBLIC SECTION.
DATA: ABC(30) TYPE C VALUE 'TECHNO INDIA UNIVERSITY'.
METHODS: T1, T2.
PROTECTED SECTION.
PRIVATE SECTION.
METHODS: M1PRIV.
METHODS: M2PRO.
ENDCLASS.

CLASS MCA IMPLEMENTATION.
METHOD T1.
WRITE: /'THIS IS T1 METHOD'.
CALL METHOD M1PRIV.
ENDMETHOD.

METHOD T2.
WRITE: /'THIS IS T2 METHOD'.
CALL METHOD M2PRO.
ENDMETHOD.
METHOD M1PRIV.
WRITE: 'THIS IS M1PRIV FROM PRIVATE SECTION '.
ENDMETHOD.
```

```
METHOD M2PRO.
WRITE: 'THIS IS M2PRO METHOD FROM PROTECTED SECTION'.
ENDMETHOD.

ENDCLASS.

START-OF-SELECTION.
DATA: OBJ TYPE REF TO MCA.
CREATE OBJECT: OBJ.
WRITE: OBJ->ABC.
CALL METHOD: OBJ->T1, OBJ->T2.
```

EXPLAINATION

This ABAP (Advanced Business Application Programming) code defines and implements a class named MCA. Let's break down the code:

1. **Class Definition (CLASS MCA DEFINITION):**
    - **PUBLIC SECTION: This section contains the public components of the class. In this case, there is a data member ABC of type character with a length of 30, initialized with the value 'TECHNO INDIA UNIVERSITY'. Additionally, there are two public methods (T1 and T2).**
    - **PROTECTED SECTION: This section contains protected components. In this example, there are no components in the protected section.**
    - **PRIVATE SECTION: This section contains private components. There is a private method M1PRIV and a private method M2PRO.**
2. **Class Implementation (CLASS MCA IMPLEMENTATION):**
    - **METHOD T1: This method is defined to write a message ('THIS IS T1 METHOD') and then call the private method M1PRIV.**
    - **METHOD T2: This method is defined to write a message ('THIS IS T2 METHOD') and then call the protected method M2PRO.**
    - **METHOD M1PRIV: This private method writes a message ('THIS IS M1PRIV FROM PRIVATE SECTION').**
    - **METHOD M2PRO: This protected method writes a message ('THIS IS M2PRO METHOD FROM PROTECTED SECTION').**
3. **Class Instantiation (START-OF-SELECTION):**

- **An object OBJ of type MCA is created using the CREATE OBJECT statement.**
- **The program then writes the value of the ABC data member of the instantiated object (OBJ->ABC).**
- **Finally, it calls the public methods T1 and T2 on the object OBJ. These methods, in turn, execute the corresponding logic and call the private and protected methods.**

In summary, this ABAP program defines a class with public, private, and protected components. It creates an object of that class, accesses a public data member, and calls public methods that, in turn, call private and protected methods. The purpose and functionality of the methods are limited to writing messages in this example.

SINGLE INHERITANCE

```
class Test1 Definition.
 Public Section.
  Methods: T1, T2.
Endclass.

class Test2 definition inheriting from Test1.
 Public Section.
  Methods: T3,T4.
Endclass.

class Test1 Implementation.
 Method T1.
  Write: ' This is T1 method from Test1'.
 EndMethod.

 Method T2.
  Write: ' This is T2 method from Test1'.
 EndMethod.
Endclass.

class Test2 Implementation.
 Method T3.
```

```
  Write: /'This is T3 method from Test1'.

  Call Method T1.

 EndMethod.


 Method T4.

  Write: /'This is T4 method from Test1'.

  Call Method T2.

 EndMethod.

Endclass.


Start-of-Selection.

Data: obj type ref to Test2.

Create object:obj.

call method: obj->T3.

call method: obj->T4.
```

EXPLAINATION

This ABAP (Advanced Business Application Programming) code defines and implements two classes, Test1 and Test2. Test2 inherits from Test1. Let's break down the code:

1. **Test1 Class Definition (class Test1 Definition):**
    - **PUBLIC SECTION: This section contains the public components of the class. In this case, there are two public methods (T1 and T2).**
2. **Test2 Class Definition (class Test2 definition inheriting from Test1):**
    - **PUBLIC SECTION: This section inherits the public components of Test1 and adds two more public methods (T3 and T4).**
3. **Test1 Class Implementation (class Test1 Implementation):**
    - **Method T1: This method is implemented to write a message ('This is T1 method from Test1').**
    - **Method T2: This method is implemented to write a message ('This is T2 method from Test1').**
4. **Test2 Class Implementation (class Test2 Implementation):**

- **Method T3: This method is implemented to write a message ('This is T3 method from Test1') and then call T1 method of the base class (Test1) using Call Method T1.**
- **Method T4: This method is implemented to write a message ('This is T4 method from Test1') and then call T2 method of the base class (Test1) using Call Method T2.**

5. **Object Instantiation and Method Calls (Start-of-Selection):**
   - **An object obj of type Test2 is created using the CREATE OBJECT statement.**
   - **The program then calls the methods T3 and T4 on the object obj. These methods, in turn, execute the corresponding logic from Test2 and call methods from the base class Test1.**

In summary, this ABAP program demonstrates class inheritance. Test1 serves as the base class, and Test2 inherits from it. Test2 introduces additional methods (T3 and T4) and implements them by calling methods from the base class (T1 and T2). The program creates an object of Test2 and calls methods on it, showcasing the inheritance and method calling behavior.

CALCULATOR

```
parameters: num1 type i,

        num2 type i,

        ch type c.

data res type i.

IF ch = '+'.

  res = num1 + num2.


ELSEIF ch = '*'.

  res = num1 * num2.


ELSEIF ch = '-'.

  res = num1 - num2.


endif.

write: 'The Result=',res.
```

RADIO BUTTON

```
Selection-screen begin of block blk1 with frame title text-111.
```

```
parameters: Male radiobutton group g11,
            Female radiobutton group g11,
            Others radiobutton group g11.
if male eq 'X'.
 write: 'MALE'.
elseif female eq 'X'.
 write 'FEMALE'.
else.
 write:'OTHERS'.
endif.



selection-screen end of block blk1.
```

EXPLAINATION

This ABAP code defines a selection screen block (blk1) with radio buttons for gender selection. Let's break down the code:

1. **Selection Screen Block (Selection-screen begin of block blk1 with frame title text-111):**
   - **blk1: This is the name of the block.**
   - **with frame title text-111: This specifies that the block should have a frame with the title 'text-111'. The title is a placeholder, and you can replace it with an actual text element.**
2. **Radio Buttons (parameters: Male radiobutton group g11, Female radiobutton group g11, Others radiobutton group g11):**
   - **Three radio buttons (Male, Female, and Others) are defined within the block. They are all part of the same radio button group (g11). This means that only one radio button can be selected at a time within this group.**
3. **Conditional Statements (if male eq 'X'. ... endif):**
   - **The code checks the selected radio button using the variables male, female, and others. These variables correspond to the three radio buttons in the group (g11).**
   - **If the Male radio button is selected (male eq 'X'), it writes 'MALE' to the output.**
   - **If the Female radio button is selected (female eq 'X'), it writes 'FEMALE' to the output.**
   - **If neither the Male nor Female radio button is selected (i.e., the else part), it writes 'OTHERS' to the output.**

4. **Selection Screen Block End (selection-screen end of block blk1):**
    - **This marks the end of the selection screen block.**

In summary, this code defines a selection screen block with three radio buttons for gender selection. The user can choose between 'Male', 'Female', or 'Others'. The selected option is determined using conditional statements, and based on the selection, the program writes the corresponding gender label to the output. The use of radio buttons ensures that only one option in the group can be selected at a time.

CHECKBOX

```
Selection-screen begin of block blk2 with frame title text-161.

parameters: java as checkbox,

      python as checkbox,

      c_prog as checkbox.


selection-screen end of block blk2.
```

EXPLAINATION

This ABAP code defines a selection screen block (**blk2**) with three checkboxes for programming language selection. Let's break down the code:

1. **Selection Screen Block (Selection-screen begin of block blk2 with frame title text-161):**
    - **blk2**: This is the name of the block.
    - **with frame title text-161**: This specifies that the block should have a frame with the title 'text-161'. The title is a placeholder, and you can replace it with an actual text element.
2. **Checkboxes (parameters: java as checkbox, python as checkbox, c_prog as checkbox):**
    - Three checkboxes (**java**, **python**, and **c_prog**) are defined within the block. Each checkbox represents a programming language (Java, Python, and C).
3. **Selection Screen Block End (selection-screen end of block blk2):**
    - This marks the end of the selection screen block.