Tutorial 58 - Abstract Base Class & Pure Virtual Functions in C++

Pure Virtual Functions in C++

- A **pure virtual function** is a function that has no implementation and is declared by assigning the value `0` to it.
- It is used to define an **abstract class**.
- Syntax:

  ```
  virtual void functionName() = 0;
  ```

Abstract Base Class in C++

- An **abstract base class** contains at least one pure virtual function.
- Any derived class must override all pure virtual functions, or the compiler will throw an error.
- The purpose of an abstract class is to provide a blueprint for derived classes.

Code Example

Code Snippet 1: Abstract Base Class `CWH`

```
class CWH {
protected:
    string title;
    float rating;
public:
    CWH(string s, float r) : title(s), rating(r) {}
    virtual void display() = 0; // Pure virtual function
};
```

- **Class** `CWH`:
  - Contains protected members `title` (string) and `rating` (float).
  - **Pure virtual function** `display()`, making `CWH` an **abstract class**.
  - Derived classes must implement `display()`.

Code Snippet 2: Derived Class `CWHVideo`

```
class CWHVideo : public CWH {
    float videoLength;
public:
    CWHVideo(string s, float r, float vl) : CWH(s, r), videoLength(vl) {}
    void display() {
        cout << "This is an amazing video with title " << title << endl;
        cout << "Ratings: " << rating << " out of 5 stars" << endl;
        cout << "Length of this video is: " << videoLength << " minutes" << endl;
    }
};
```

- **Class** `CWHVideo`:

  - Inherits `CWH` and implements the `display()` function.
  - Has an additional member `videoLength` to represent the video duration.

Code Snippet 3: Derived Class `CWHText`

```
class CWHText : public CWH {
    int words;
public:
    CWHText(string s, float r, int wc) : CWH(s, r), words(wc) {}
    void display() {
        cout << "This is an amazing text tutorial with title " << title << endl;
        cout << "Ratings of this text tutorial: " << rating << " out of 5 stars" << endl;
        cout << "No of words in this text tutorial is: " << words << " words" << endl;
    }
};

```

- **Class** `CWHText`:
  - Inherits `CWH` and implements the `display()` function.
  - Has an additional member `words` to represent the number of words in the text.

Code Snippet 4: Main Program

```
int main() {
    string title;
    float rating, vlen;
    int words;

    // For CWHVideo
    title = "Django tutorial";
    vlen = 4.56;
    rating = 4.89;
    CWHVideo djVideo(title, rating, vlen);

    // For CWHText
    title = "Django tutorial Text";
    words = 433;
    rating = 4.19;
    CWHText djText(title, rating, words);

    // Array of pointers to base class
    CWH* tuts[2];
    tuts[0] = &djVideo;
    tuts[1] = &djText;

    tuts[0]->display();
    tuts[1]->display();

    return 0;
}

```

- **Main Program**:
  - Creates objects of `CWHVideo` and `CWHText`.
  - Uses an array of base class pointers (`CWH*`) to point to the derived class objects.

  - Calls the overridden `display()` function through the base class pointers.

Key Points

1. **Pure Virtual Function**:
   - Declared by `= 0` in the base class.
   - Makes the base class **abstract**.
   - Derived classes **must** override this function, or the compiler will throw an error.
2. **Abstract Base Class**:
   - Contains at least one pure virtual function.
   - Cannot create objects of an abstract class.
3. **Derived Classes**:
   - Must implement the pure virtual function to become concrete classes.
   - Can have their own additional members and methods.

Error Handling

- If the derived class does not implement the pure virtual function, the compiler will show an error indicating that the function is not overridden.

Short Notes for Notebook

Pure Virtual Function:

- A function with no implementation.
- Syntax: `virtual void functionName() = 0;`.
- Used to define **abstract classes**.
- Must be overridden in derived classes.

Abstract Base Class:

- A class with at least one pure virtual function.
- Derived classes must provide an implementation for all pure virtual functions.
- Cannot create objects of an abstract class.

Example:

1. **Base Class (`CWH`)**:
   - Contains a pure virtual function `display()`.
2. **Derived Classes (`CWHVideo`, `CWHText`)**:
   - Override `display()` and provide specific functionality.

Key Concept:

- **Abstract class** = Class with at least one **pure virtual function**.
- Derived classes must **override** the pure virtual function to be instantiated.

Example Output:

```
This is an amazing video with title Django tutorial
Ratings: 4.89 out of 5 stars
```

```
Length of this video is: 4.56 minutes

This is an amazing text tutorial with title Django tutorial Text
Ratings of this text tutorial: 4.19 out of 5 stars
No of words in this text tutorial is: 433 words

```