

Basic Software Engineering

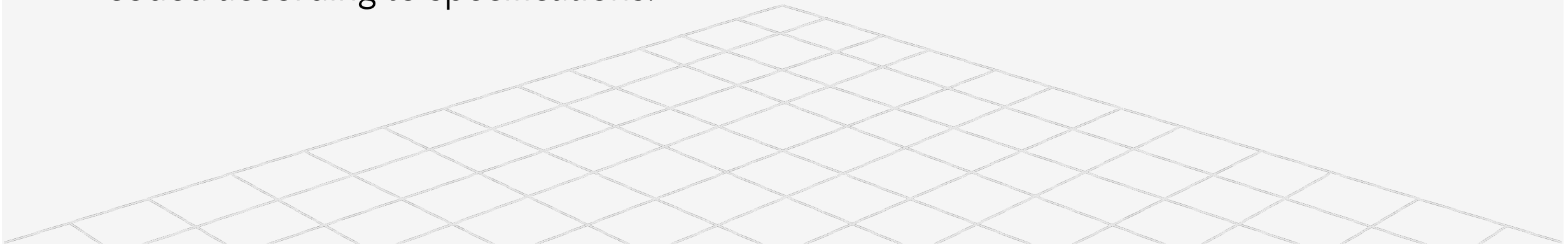
Unit – 6 Coding Approaches

By: Vibhuti Patel

PICA – BCA

Coding

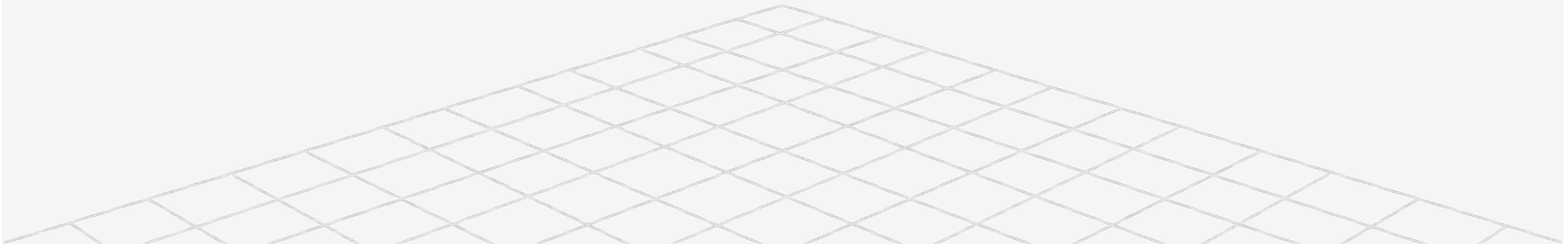
- Coding is undertaken once design phase is complete & design documents have been successfully reviewed.
- In the coding phase, every module identified and specified in the design document is independently coded & unit tested.
- The input to the coding phase is the design document.
- During the coding phase, different modules identified in the design document are coded according to specifications.



Coding

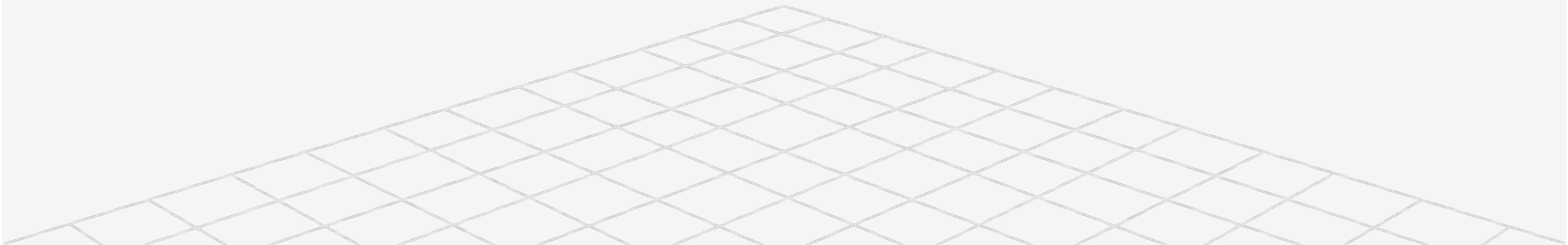
Most software development organizations formulate their own coding standards that suit them most & require their engineers to follow these standards rigorously due to the following reasons:

- A coding standard gives a uniform appearance to the codes
- It provides better understanding of the code
- It encourages good programming practice



Programming Practice

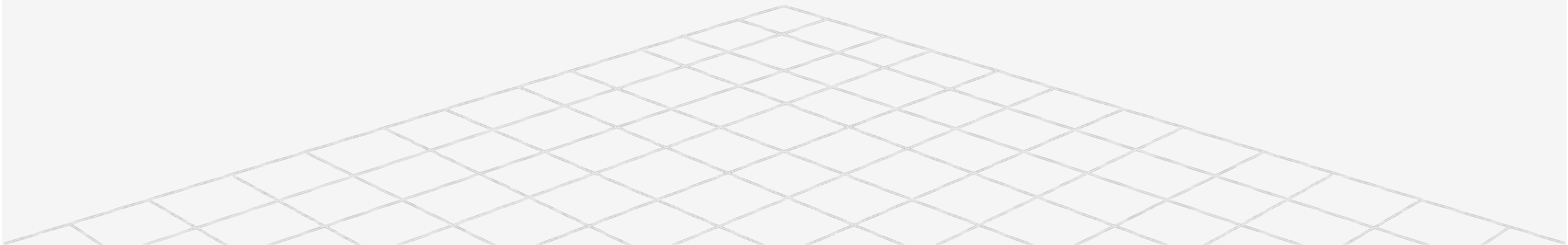
- The primary goal of the coding phase is to translate the given design into source code in a given programming language, so that the code is simple easy to test and easy to understand and modify.
- So now we will discuss some concepts related to coding in a language independent manner.



Programming Practice

Top-down & Bottom-up: All design contains hierarchies as creating a hierarchy is a natural way to manage complexity.

- In a Top down implementation, the implementation starts from the top of the hierarchy and proceeds to the lower levels.
- In a Bottom-up implementation the process is the reverse.



Programming Practice

Top-down

- A top-down approach is essentially breaking down of a program into small programs (or modules) in a reverse engineering fashion.
- Structure / procedure oriented programming languages like C programming language follows top-down approach.
- In top-down approach, main function is written first and all sub functions are called from main function.

Bottom-up

- A bottom-up approach is piecing together of module (or small program) to give rise to more complex program.
- Object oriented programming languages like C++ and JAVA programming language follows bottom-up approach.
- In bottom-up approach, code is developed from modules and then these modules are integrated with main function.

Programming Practice

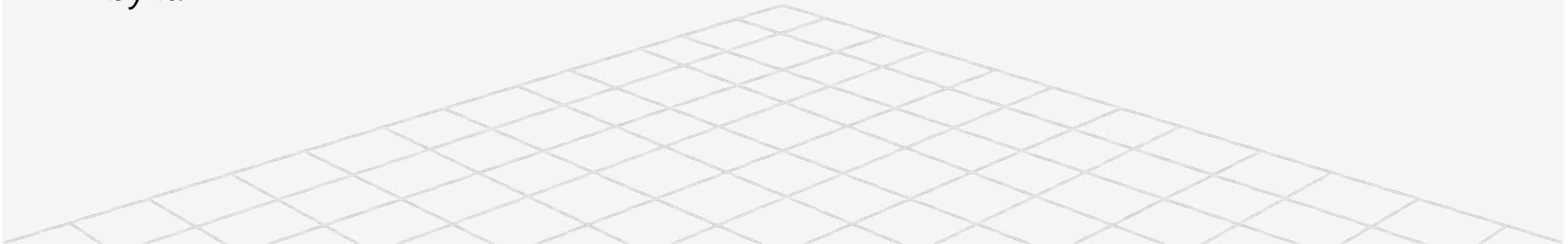
Structured Programming:

- It is also known as **goto-less** programming.
- A program has a static structure as well as dynamic structure.
- The static structure is the structure of the text of the program, which is usually just a linear organization of statements of the program.
- The dynamic structure of the program is the sequence of statements executed during the execution of the program.
- The goal of structured programming is to ensure that the static structure and the dynamic structures are the same.

Programming Practice

Information Hiding:

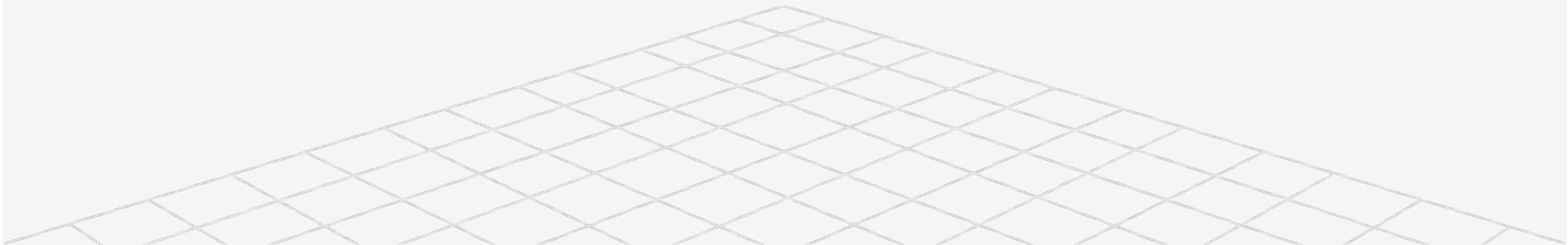
- Information hiding can reduce the coupling between modules and make the system more maintainable.
- Information hiding is also an effective tool for managing the complexity of developing software.
- Another form of information hiding is to let a module see only those data items needed by it.



Programming Practice

Programming Style:

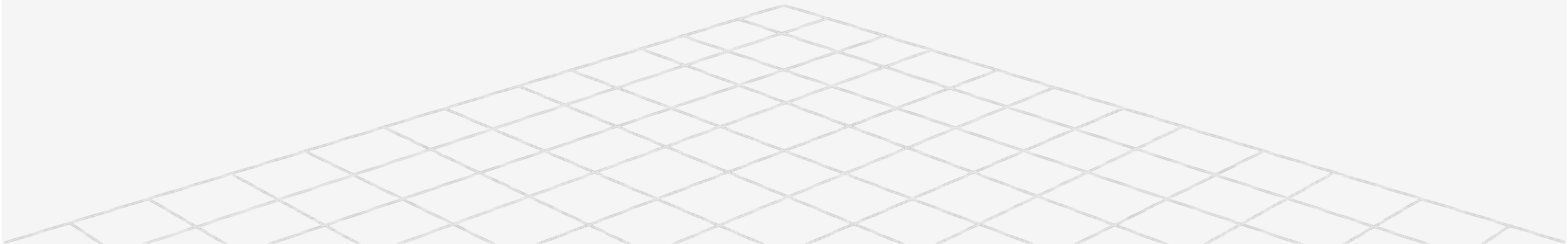
- The programming style consists of some standard and guidelines which we will discuss in the next phase of this presentation.



Programming Practice

Internal Documentation:

- Internal documentation of the program is done by the use of comments.
- All languages provide a means for writing comments in program.
- Comments are textual statements that are meant for the program reader and are not executed.



Coding Standards

Indentation: It is an empty space at the beginning of a line to signal the start of a new paragraph.

Proper & consistent indentation is important in producing easy to read & maintainable programs.

```
while (x == y)
{
    something();
    somethingelse();
}
```

Coding Standards

Inline Comments: Inline comments explaining the functioning of the subroutine.

```
/**
 * Simple HelloButton() method.
 * @version 1.0
 * @author john doe <doe.j@example.com>
 */
HelloButton()
{
    JButton hello = new JButton( "Hello, wor
    hello.addActionListener( new HelloBtnList

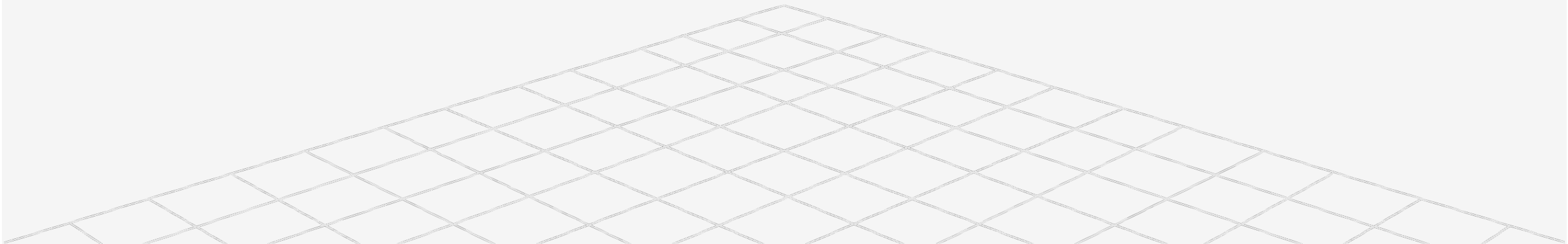
    // use the JFrame type until support for t
    // new component is finished
    JFrame frame = new JFrame( "Hello Button"
    Container pane = frame.getContentPane();
    pane.add( hello );
    frame.pack();
    frame.show();           // display the fra
}
```

An illustration of Java source code with **prologue** comments indicated in red and **inline** comments in green. Program code is in blue.

Coding Standards

Structured Programming:

- It is also known as **goto-less** programming.
- A program has a static structure as well as dynamic structure.
- The goal of structured programming is to ensure that the static structure and the dynamic structures are the same.



Coding Standards

Classes, functions or methods: Keep classes, functions or methods reasonably sized. This depends upon the language being used.

- A good rule for module length is to constrain each module to one function or action (i.e. each module should only do one “thing”).
- The names of the classes, functions or methods shall have verbs in them. That is the names shall specify an action,

e.g. “get_name”

“compute_temperature”



Coding Standards

Source Files: The name of the source file or script shall represent its function.

e.g. file name for arithmetic operations program file

arithmetic.cpp

Variable Names: Variable shall have meaningful names that convey to a casual observer, the intent of its use. Variable must be declared before they are used.

e.g. variable declaration for summation of two numbers & store it in variable

int no1, no2, answer;

Coding Standards

Use of Braces:

Bad:

```
if (j == 0)
    printf ("j is zero.\n");
```

Better:

```
if (j == 0)
{
    printf ("j is zero.\n");
}
```


Coding Standards

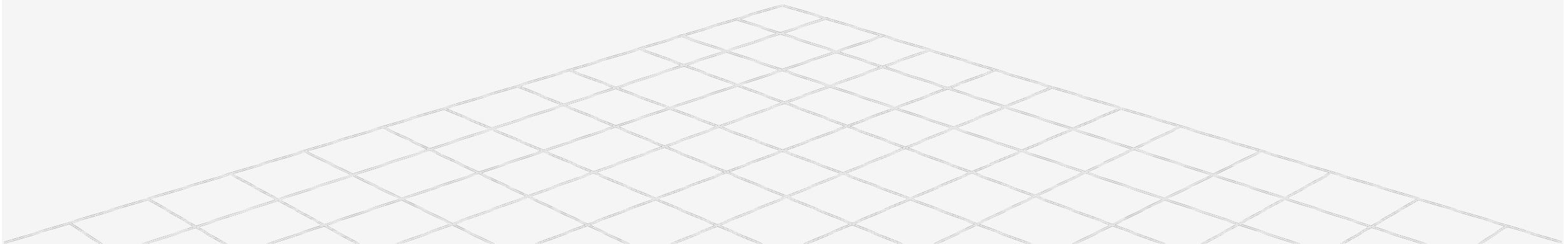
Compiler Warnings:

- Compilers often issue two types of messages: warnings and errors.
 - **Compiler warnings** normally do not stop the compilation process.
 - **Compiler errors** do stop the compilation process, forcing the developer to fix the problem and recompile.
-
- **Compiler warnings shall be treated as errors and fixed.**

Coding Guidelines

- General coding guidelines provide the programmer with a set of best practices which can be used to make programs easier to read and maintain.
- Most of the examples use the C language syntax but the guidelines can be applied to all languages.

Line Length: It is considered good practice to keep the lengths of source code lines at or below 80 characters. Lines longer than this may not be displayed properly.



Coding Guidelines

Spacing: The proper use of spaces within a line of code can enhance readability.

Bad

```
ans=no1+no2;
```

Good

```
ans = no1 + no2;
```

Coding Guidelines

Wrapping Lines: When an expression will not fit on a single line, break it.

Bad:

```
longName1 = longName2 * (longName3 +  
    LongName4 -  
longName5) + 4 * longName6 ;
```

Better:

```
longName1 = longName2 * (longName3 +  
    LongName4 - LongName5)  
+ 4 * longName6 ;
```

Coding Guidelines

Variable Declarations: Variable declarations that span multiple lines should always be preceded by a type.

Acceptable:

```
int price , score ;
```

Acceptable:

```
int price ;
```

```
int score ;
```

Not Acceptable:

```
int price ,  
score ;
```

Coding Guidelines

Program Statements: It should be limited to one per line. Also, nested statements should be avoided when possible.

Bad:

```
number_of_names = names.length ; b =  
    new JButton [ number_of_names ] ;
```

Better:

```
number_of_names = names.length ;  
b = new JButton [ number_of_names ] ;
```

Coding Guidelines

Use of parenthesis: It is better to use parentheses liberally.

Acceptable:

```
total = 3 - 4 * 3 ;
```

Better:

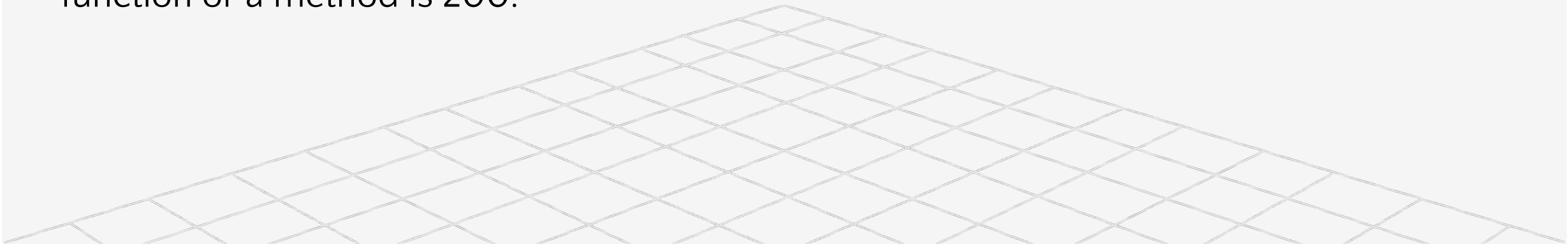
```
total = 3 - ( 4 * 3 ) ;
```

Coding Guidelines

Inline Comments: Inline comments promote program readability.

Meaningful Error Messages: Use error handling concepts. It handle errors but also involves making error messages meaningful.

Reasonably Sized Functions or Methods: A reasonable number of lines of code for function or a method is 200.



Code Verification

Code Reading: Code reading involves careful reading of the code by the programmer to detect any discrepancies between the design specifications and the actual implementation.

Code Inspection or Review: Review process was started with the purpose of detecting defects in the code.

Code reviews are designed to detect defects that originate during the coding process, although they can also detect defects in detailed design.

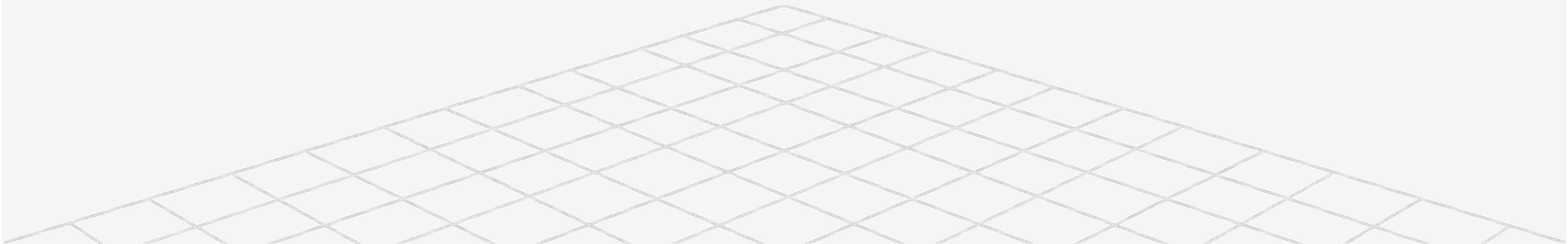


Code Verification

Unit Testing: It is a dynamic method for verification where the program is actually compiled and executed.

It is one of the most widely used methods & the coding phase is sometime called the “coding and unit testing phase”.

It is a level of software testing where individual units/ components of a software are tested. The purpose is to validate that each unit of the software performs as designed.



Thank You...

