Good written code

Use indentation to highlight the structure of code

Some important best practices for writing readable code:

1 - Commenting & Documentation

The comment should disclose things that can not be immediately learned from the code.

2 - Consistent Indentation

Style 2:

```
01  function foo()
02  {
03      if ($maybe)
04      {
05          do_it_now();
06          again();
07      }
08      else
09      {
10          abort_mission();
11      }
12      finalize();
13  }
```

Style 3:

```
01  function foo()
02  {   if ($maybe)
03      {   do_it_now();
04          again();
05      }
06      else
07      {   abort_mission();
08      }
09      finalize();
10  }
```

Style 1:

```
1   function foo() {
2       if ($maybe) {
3           do_it_now();
4           again();
5       } else {
6           abort_mission();
```

But that is only a matter of preference. There is no "best" style that everyone should be following. Actually, the best style, is a consistent style. If you are part of a team or if you are contributing code to a project, you should follow the existing style that is being used in that project.

3. Use Descriptive Names

When you use unclear and non-descript names for variables, classes, and functions, **you're essentially obfuscating the application logic from any programmer who reads the code**, including yourself.

What does a variable named dxy actually mean? Who knows. You'd probably have to read the entire chunk of code to reverse engineer its meaning. On the other hand, the meaning of a variable like distanceBetweenXY is instantly recognizable.

But, this doesn't necessarily apply to temporary variables. They can be as short as a single character.

mysql_real_escape_string()

MysqlRealEscapeString()

Consistent Naming Scheme

4 - Avoid Obvious Comments

```php
// get the country code
$country_code = get_country_code($_SERVER['REMOTE_ADDR']);

// if country code is US
if ($country_code == 'US') {

    // display the form input for state
    echo form_input_state();
}
```

```php
// display state selection for US users
$country_code = get_country_code($_SERVER['REMOTE_ADDR']);
if ($country_code == 'US') {
    echo form_input_state();
}
```

5 - Code Grouping

```
01  // get list of forums
02  $forums = array();
03  $r = mysql_query("SELECT id, name, description FROM forums");
04  while ($d = mysql_fetch_assoc($r)) {
05      $forums []= $d;
06  }
07
08  // load the templates
09  load_template('header');
10  load_template('forum_list',$forums);
11  load_template('footer');
```

6 - DRY Principle

DRY stands for Don't Repeat Yourself. Also known as DIE: Duplication is Evil.

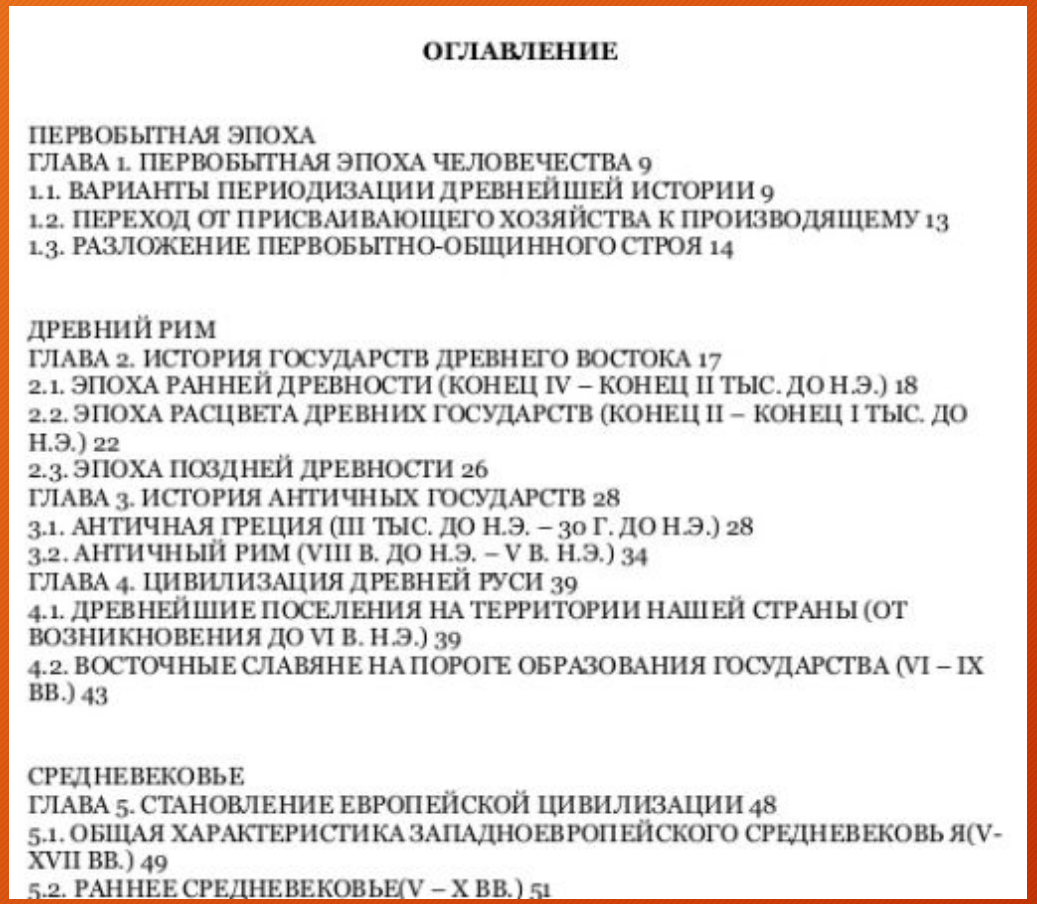
7 - Avoid Deep Nesting

Too many levels of nesting can make code harder to read and follow.

```
01  function do_stuff() {
02
03  // ...
04
05      if (is_writable($folder)) {
06
07          if ($fp = fopen($file_path,'w')) {
08
09              if ($stuff = get_some_stuff()) {
10
11                  if (fwrite($fp,$stuff)) {
12
13                      // ...
14
15                  } else {
16                      return false;
17                  }
18              } else {
19                  return false;
20              }
21          } else {
22              return false;
23          }
24      } else {
25          return false;
26      }
27  }
```

Functions and procedures

Computer programs can consist of thousands of lines of code, just like a textbook can have thousands of words.
In the same way that a textbook is divided into chapters, a program is divided into related functionality using **modules**.



ОГЛАВЛЕНИЕ


```
protected override void Dispose(bool disposing)
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}

#region Windows Form Designer generated code

/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
private void InitializeComponent()
{
    this.SuspendLayout();
    //
    // Form1
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
    this.ClientSize = new System.Drawing.Size(395, 278);
    this.Name = "Form1";
    this.Text = "Form1";
    this.Click += new System.EventHandler(this.Form1_Click);
    this.ResumeLayout(false);

}
```

The C # program is built from modules, the role of which is performed by *classes*.
The functionality of the class is provided by methods.

In C #, these methods are of two kinds: procedures and functions.

The syntax for declaring a method allows you to uniquely determine what the method is, a procedure or a function.

<access modifier> <type> <name> (parameters)
{
Body functions
Return <type>;
}

<access modifier> void <name> (parameters)
{
body procedures
}

What is the class name?
How many methods does the class include?
What are the names of these methods?
What access modifier?
What kind of methods, procedures or functions?
What events correspond to the methods?
What actions are performed when these methods are called?
What parameters?

Example

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        label1.Text = "Hello!";
    }

    private void button2_Click(object sender, EventArgs e)
    {
        label1.Text = "Good bye!";
    }
}
```

Global variable - declared at the start of the program, their global *scope* means they can be used in any procedure or subroutine in the program

Local variable declared within subroutines or programming blocks, their local scope means they can only be used within the subroutine or program block they were declared in

```
<access modifier> <type> <name> (parameters)
{
Body functions
Return <type>;
}
```

```
<access modifier> void <name> (parameters)
{
body procedures
}
```

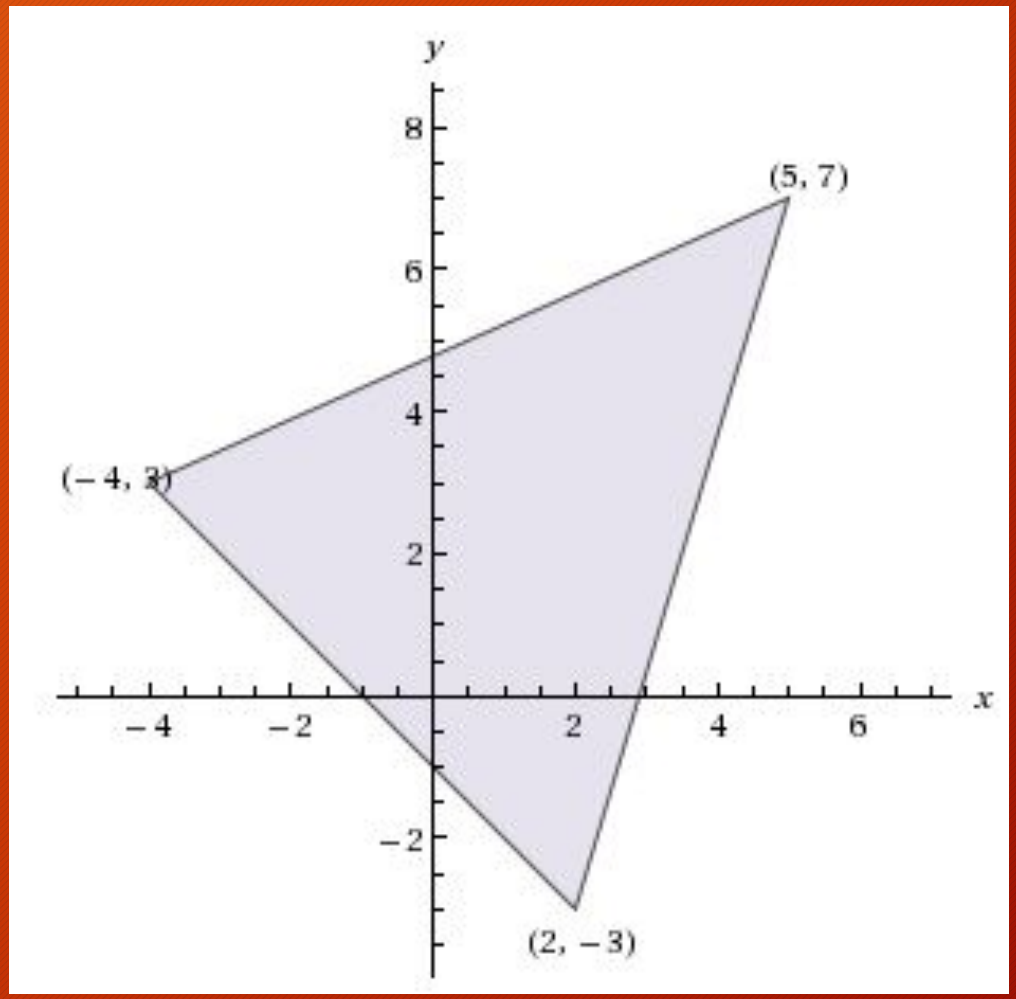
What are the differences?

Procedure	Function
returns a formal result void indicating that there is no result	always computes a value returned as the result of the function
call procedure is the operator of the language	is called in expressions

What are the procedures and functions for?

- They help you structure your code
- They allow you to create a common routine once and re-use as many times as you want
- They allow you to share code with other programs
- They allow you to test sub routines independently of the rest of the code

Draw a triangle and calculate its area



1 What is the difference between a procedure and a function?

- There is no difference
- A procedure performs a task whereas a function produces information
- A procedure produces information whereas a function performs a task

2 What is a module?

- A module is a small section of a program
- A module allows you to pass information into a function
- A module is the name for a value passed into a function

3 What is a parameter?

- Parameters are the values that are passed into a function or procedure
- Parameters are the names of the information that is used in a function or procedure
- A parameter is a small section of program

4 What is an argument?

- An argument is a value that is passed into a function or procedure
- An argument is a small section of program
- An argument is the name of the information that is used in a function or procedure

5 What is the difference between a local and a global variable?

- A global variable is declared so it can be used in any part of the program and a local variable is only available in a specific function
- A local variable is declared so it can be used in any part of the program and a global variable is only available in a specific function
- Both local and global variables can be used in any part of a program

6 What is a module-scoped variable?

- A variable that is available in any part of the program
- A variable that can only be used once
- A variable that is only available in a single file or module

7 What is meant by 'executing' a function?

- Deleting the function
- Changing the function
- Asking a function to run

8 Which of the following can be given an identifier?

- Functions and procedures
- Functions, procedures, parameters, arguments, variables and constants
- Variables and constants

9 In the statement 'for x in range (1,5)' what type of variable is 'x'?

- Local variable
- Global variable
- Module-scoped variable

10 Which of the following is a function?

- Algorithm to draw a circle
- Algorithm to calculate VAT on a product
- Algorithm to create a cake