**1.** What is the difference between a declaration and a definition?

With declaration we simply state that somewhere in the program exists such variable or a function. By defining the variable or function we allocate the memory for the object and specify what the object have to do (e.g. a value of the variable, or a piece of code in function)

**2.** How do we syntactically distinguish between a function declaration and a function definition?

We specify only the name of function, type of return value and at least type of the arguments, if we declare the function. When we define it, we must specify names of arguments and the code. There can be more than one declaration, but there have to be only one definition in program.

**3.** How do we syntactically distinguish between a variable declaration and a variable definition?

When we provide the type of the variable or type and the value we define it. That means the memory for this variable has been allocated. When we use keyword **extern** no memory have been used, but we have stated, that somewhere in other place of program this variable was defined.

**4.** Why can’t you use the functions in the calculator program from Chapter 6 without declaring them first?

No matter how we change the order of them, one of it still requires another one to be declared or defined earlier.

**5.** Is int a; a definition or just a declaration?

Both definition and declaration, because the memory for this object has been already allocated. And definition can be also declaration

**6.** Why is it a good idea to initialize variables as they are declared?

Not-initialized variables can cause hardware errors.

**7.** What can a function declaration consist of?

Type of return value (or void, if we don’t need to return the value), list of parameters, reference symbol (&), const keyword (if we use pass-by-const method of providing arguments), and the code. When we just declare the function, we are allowed not to specify the name of arguments and the code.

**8.** What good does indentation do?

It improves readability of code and therefore makes it less error-prone.

**9.** What are header files used for?

Header files hold declarations of objects, that are defined in some file, but have to be used in the another one.

**10.** What is the scope of a declaration?

It is an area of program where particular object is declared and can be used till the end of the scope.

**11.** What kinds of scope are there? Give an example of each.

The global scope - holds all other possible scopes

A namespace scope - a named scope nested in the global scope or in another namespace

A class scope - the area of text within a class

A local scope - between { . . . } braces of a block or in a function argument list

A statement scope - While, for, if, switch etc. statements

**12.** What is the difference between a class scope and local scope?

A local scope can be nested in class scope. All values defined within the class scope can be used in local scope of this class but not in another direction.

**13.** Why should a programmer minimize the number of global variables?

Because there is no method to track what function changes global variable.

**14.** What is the difference between pass-by-value and pass-by-reference?

When we use pass-by-value method, the value of argument is copied to the new variable and after the function has done all the work, this variable is destroyed. Pass-by-reference method allows us to change the value of provided argument by simply creating a reference on it.

**15.** What is the difference between pass-by-reference and pass-by-const-reference?

Using pass-by-const-reference the function doesn’t change the value of the argument.

**16.** What is a swap()?

A function from standart library that swaps the values of provided arguments.

**17.** Would you ever define a function with a vector<double>-by-value parameter?

A better way to use vector as a parameter is to use pass-by-const-reference, so that we won’t create unnecessary vector and copy values of another vector in it. Also, the function can’t override values of provided argument because of keyword **const**.

**18.** Give an example of undefined order of evaluation. Why can undefined order of evaluation be a problem?

Because we don’t know for sure, if the value is calculated in left part or in right part first.

Examples: v[i] = ++i;

v[++i] = i;

a = ++i + ++i;

cout << ++i << ++ i;

**19.** What do x&&y and x||y, respectively, mean?

x&&y means X and Y, or in this case “if x and y are true”

x||y - x or y, or “if x is true or if y is true”

**20.** Which of the following is standard-conforming C++: functions within functions, functions within classes, classes within classes, classes within functions?

Functions within classes - member functions

Classes within classes - nested classes

**21.** What goes into an activation record?

List of function parameters

Address to return the value

Space for the local variables

**22.** What is a call stack and why do we need one?

It is an order in which functions are being executed during the run of the program. It operates according to the rule “Last in, first out”.

**23.** What is the purpose of a namespace?

We use namespaces to group classes, functions, data and types into an identifiable and named part of program without a type.

**24.** How does a namespace differ from a class?

Namespace doesn’t provide a type.

**25.** What is a using declaration?

“Using” declaration with object’s name means that we want to use a member of the namespace specified with “using”.

**26.** Why should you avoid using directives in a header?

It is a potential source of errors, because we can loose track of which names come where.

**27.** What is namespace std?

It is a namespace of standart library.