

INTRODUCTION TO PROGRAMMING

ASSIGNMENT 1

Day 1

1. Give 3 reasons why C is the best choice for a programming language.

- ans) → It is memory efficient - used widely in embedded systems, and is portable.
 → Fast runtime and compile times.
 → Gives a good understanding about the lower level functioning of data structures, which can be applied to higher level languages. It is a powerful language.

2. What does the compiler do?

- ans) The compiler acts as a 'translator' of sorts that translates source code from one language to another. In C, the compiler translates converts code written in C into machine language (binary), for instance, in the output file.

3. What are the steps in the program development cycle?

ans) The following are the steps:

- > Writing the source code using an editor
- > Compiling using a compiler
- > Link the program using a linker
- > Executing the program and debugging

4. What command do you need to enter in order to compile a program called 'program1.c' with your compiler?

ans) > gcc -o program1.out program1.c

5. Does your compiler do both the linking and compiling with just one command, or do you have to enter ~~separate~~ separate commands?

ans) Here, it is done in the same command for program1.c.

6. What extension should you use for your C source files?
- ans) '.c' is the extension that should be used.
7. Is 'FILENAME.TXT' a valid name for a C source file?
- ans) No, it will give a 'file format not recognized' error.
8. If you execute a program that you have compiled and it doesn't work as you expected, what should you do?
- ans) If it doesn't work as expected, then one should debug it and modify the code, if necessary.
9. What is a machine language?
- ans) A machine language is the language that the computer understands - it is in binary instructions that the machine understands and executes.
10. What does the linker do?
- ans) The linker combines the object code from the source code from the functions library, and creates an executable.

Day 2

1. What is the term for a group of one or more C statements enclosed in braces?
- ans) It is called a ~~code~~ 'code block'.
2. What is one component that must be present in every C program?
- ans) The main() function must be present in every C program.
3. How do you add program comments, and how are they used?

ans) Multi-line program comments in C are added with '`/*`' and '`*/`', where all code in between these are not executed. Single-line comments are made with ~~#~~ prefixing the text on the line with '`//`'. It is good practice to use program comments to make mentions and notations on what that piece of code is about, or how it operates. It can also be useful, since it can provide context if the code is to be modified after a while. ~~example~~

4. What is a function?

ans) A function is a section of the code that may take ~~pro~~ parameters (if needed), and performs a specific set task or set of tasks assigned to it when called. ~~at 12~~

5. C offers two types of functions. What are they, and how are they different?

ans) C offers library, and user-defined functions. The former comes with the compiler, and the latter is created by the user / programmer.

6. What is the `#include` directive used for?

ans) `#include` acts as a 'substitute' or sorts, since during compilation, it instructs the compiler to add code from the file mentioned in the `#include` onto the source code.

7. Can comments be nested?

ans) While it isn't good practice to nest comments (also for readability), some compilers do allow this, while others don't.

8. Can comments be longer than one line?

ans) Yes, multi-line comments are possible by prefixing the code with '`/*`' and suffixing it with '`*/`', and anything in between will be ignored.

9. What is another name for an include file?

ans) An include file is also known as a header file.

10. What is an include file?

ans) It is an independent disk file that is needed by the compiler to use different functions, and contains information needed for that.

Day 3

1. What's the difference between an integer variable and a floating point variable?

ans) An integer variable holds a ~~real~~ number that is not a fraction (i.e. it can be +ve or -ve, but there's no fractional part attached to it), while a floating point variable can hold any real number (can also have a fractional part).

2. Give two reasons for using a double-precision floating point variable (type `double`) instead of a single-precision floating-point variable (type `float`).

ans) A double type variable is preferred over a floating-point type variable for the following reasons:

> can

> it can hold a bigger range of values than float

> it is more precise than float.

3. What are five rules that you know are always true when allocating size for variables?

ans) > size of an int \leq size of a type ~~than~~ long

> size of a char is one byte

> size of a short \leq size of an int

> size of a float \leq size of a double

> size of an int is equal to the size of an unsigned

4. What characters are allowed in C variable names?

ans) Underscores, letters and numerals are allowed. However, it cannot start with a numeral.

5. What guidelines should you follow in creating names for variables and constants?

ans) Constant names should be in uppercase and variable names must be in lowercase. The names must be named aptly, and must be self-descriptive (must describe the data it contains).

6. What's the minimum value that a type int variable can hold?

ans) If it is a signed int, -32,768 is the minimum.

If it is an unsigned int, 0 is the minimum value.

Day 4

1. What is the following C statement called, and what is its meaning?

$x = 5 + 8;$

ans) It is called an assignment statement which is instructing the computer to add 8 to 5, and then assign it to 'x'.

2. What is an expression?

ans) Anything that gives / evaluates to a numerical value is called an expression.

3. In an expression that contains multiple operators, what determines the order in which operations are performed?

ans) The operator precedence determines the order.

4. If the variable x has the value 10, what are the values of x and a after each of the following statements are executed separately?

$a = x++;$

$a = ++x;$

ans) After 1st statement : $a = 10, x = 11$

After 2nd statement : $a = 11, x = 11$

5. To what value does the expression $10 \times .3$ evaluate.

ans) It evaluates to 1.

6. To what value does the expression $5 + 3 * 8 / 2 + 2$ evaluate?

ans) It evaluates to 19.

7. Rewrite the expression in question b, adding parenthesis so that it evaluates to 16.

ans) $(5+3)*8/(2+2)$ is the required expression.

8. If an expression evaluates to false, what value does the expression have?

ans) It then has a value of null or 0.

9. In the following list, which has higher precedence?

a) == or <

b) * or +

c) != or ==

d) >= or >

ans) a) < has the higher precedence

b) * has the higher precedence

c) != and == are evaluated left ^{to} right since both have equal precedence

d) >= and > have equal precedence.

10. What are the compound binary assignment operators, and how are they useful?

ans) Examples of such are /=, -=, += etc.. They let the user combine a binary mathematical operator with an assignment operator.

Day 5

1. Will you use structured programming when writing C programs?
- ans) Yes, since that is good practice; and is convenient.
2. How does structured programming work?
- ans) It takes a complex task/problem and breaks it into multiple simpler tasks.
3. How do C functions fit into structured programming?
- ans) Separate functions can be written to perform the simpler tasks that have been broken ~~into~~ down from the complex problem and then called if necessary.
4. What must be the first line of a function definition, and what information does it contain?
- ans) The first line must be the function header that contains the name of the function, its return type, and its parameters.
5. How many values can a function return?
- ans) It can return either no value, or only one value that can be of any variable type.
- b. If a function doesn't return a value, what type should it be declared?
- ans) It must be declared of type void.
7. What's the difference between a function definition and a function prototype?
- ans) A function definition contains parameters, header, and all statements and instructions required, while the prototype only contains the header and ends with a semi-colon.

8. What is a local variable?

ans) A local variable is a variable whose scope is defined only within a function.

9. How are local variables special?

ans) Local variable ~~are~~ do not interfere, and are independent from other variables in the program.

10. Where should the main() function be placed?

ans) It must be the first function in the listing.