

48430 REVISION

BEESHANGA ABEWARDANA JAYAWICKRAMA

**UTS:
ENGINEERING AND
INFORMATION
TECHNOLOGY**

SUBJECT MINIMUM REQUIREMENTS

In order to pass the subject, a student must:

- **Achieve a minimum of 50% in the overall mark**

FINAL ASSESSMENT OVERVIEW

Final Assessment is:

FULLY OPEN BOOK

EXAM OVERVIEW

Follow ANSI C standard.

Don't include any non standard header files i.e. use `stdio.h`, `stdlib.h`, `string.h` and `math.h` only.

Follow **good programming habits (not including comments)**.

IF there is any ambiguity in any question, state your assumptions clearly. Marker will evaluate the validity of assumptions.

IN THE EXAM

You must be able to **write programs** according to a given **specification**

You must be able to **write functions** according to a given **specification**

You must be able to **explain C programming concepts** and **provide examples**

BATTLES AT THE EXAM

Applying the C programming **skills** and recalling the **knowledge** gathered

Writing code on a computer – all good programming habits will be assessed

Open book vs closed book – you will need to **T-H-I-N-K** in open book exams

NO one else can help during the exam

WHAT TO STUDY

- Lab tasks
- Lab preparation tasks/questions
- Code developed in forums and the concepts covered in them
- Forum notes
- Textbook

Programming is like driving. The more you practice the better you get at it.

Only way to pass the exam is to **practice writing code** (reading code is NOT equal to writing code).

These revision slides only summarise the important topics in the subject. They do not directly indicate the complete list of questions that will appear in the final exam and/or the weightage.

Follow the colour scheme:

- **Very important**
- Neutral / **Neutral**
- **Less important**

01 PROGRAMMING FUNDAMENTALS

Steps to follow when running a high level language program.

Software development stages – requirement, analyse, design, implement, test, maintain.

The general structure of a C program.

Pre-processors.

Variables - name/type, definition/initialisation.

Correct usage of variables when using printf() and scanf().

Types of errors – syntax, runtime and logic.

02 DATA TYPES, OPERATORS AND SIMPLE FUNCTIONS

C primitive data types – char, int, float (approx.), double (approx.).

Order of operations – Top to bottom, left to right. unary (+, -), binary (+, -, *, /, %, ()), assignment (=).

What is the value of c? `int a = 1, b = 2; float c; c = a/b;` Is the expression incorrect? **Type casting**.

Functions – pre-defined/user-defined, **prototype/definition**.

Basic structure of a function – return type, function name, input parameters, definition.

How to call a function.

Scope of variables.

03 CONTROL STRUCTURES (IF AND LOOP)

Conditional and logical operations - <, >, <=, >=, ==, !=, &&, ||, !

Order of operations – arithmetic operations, unary logical operations, binary conditional operations, binary logical operations, assignment.

e.g. `flag = !y && x * -1 + 5 <= 0`

“Short-Circuit” evaluation.

Can I write: `1 < x <= 10` ?

General structure of a **if-else** and **switch** statements. **Break** statements.

General structure of **for**, **while** and **do-while** statements. Initialisation, check condition, variation.

Counter controlled and Sentinel controlled.

04 ARRAYS AND STRINGS

How to **define an array** (size – constant), how to **access an element**, how to initialise.

Passing arrays to functions – **scope**, **size**, **const** keyword.

Consider `int a[10];` Value/address – `a`, `a[0]`, `&a[0]`. True/false – `a==&a[0]` ? Can I write - `int* p = a` ?

Dynamic sized arrays. How to define and use?

Strings in C are – `char stringa []`, or `char* stringp`.

String size is managed by – null termination (`'\0'`).

Some functions from `string.h` – **`strcpy()`**, **`strcmp()`**, **`strcat()`**.

05 STRUCTURES AND FILE PROCESSING

How to **define a structure type**, how to **access an element**, how to initialise.

Passing structures to functions – scope

Structure pointers. `planet_t *p; p = (planet_t*) calloc(1, sizeof(planet_t)); p->diameter = 12.1;`

Can a function return a struct?

Text/binary files – differences, advantages and disadvantages.

Text files – fprintf, fscanf. Binary files – fwrite, fread. How to read till the end of a file.

06 07 **POINTERS** AND DYNAMIC MEMORY MANAGEMENT

What is a pointer? An address.

Meaning of the following terms: `int x; x; &x;`

Meaning of the following terms: `int* xp; xp; *xp;`

Meaning of the following terms: `void fun(double* yp); double d; fun(&d);`

Scope of variables – block scope.

Write functions to output more than one result using pointers.

Pointers and arrays

`stdlib.h` functions – `malloc`, `free`

Common mistakes with pointers – uninitialised pointers, dangling pointers, double deallocation, memory leaks

Linked lists

09 10 11 OTHER TOPICS

Programming in the large – developing custom libraries, compiling multiple c source codes, **makefiles**

Cross development for embedded systems

Command line parsing

On to C++ (NOT EXAMINABLE)

ONLINE EXAM INSTRUCTIONS

- General online instructions on Canvas.
- Access and complete the assessment on Ed
- You have to manually Click “Mark” once each answer is given. Answer will NOT be automatically submitted.
- Final assessment includes multiple questions. Each question is individually submitted. You can move between questions. **Before moving to the next question, click “Mark” to save your answer to the current question.** You can change your answers any time before the due time of the assessment by submitting a new answer. Multiple attempts of submissions are allowed before the due time of the assessment. Only the LAST submissions before the due time will be marked. Late submissions after the due time will not be accepted.
- If close your browser or use the back/backspace button, you will lose the unsubmitted answers and if you re-enter to Ed, you will only see the last previously submitted answer.
- If you wish to reset the question scaffold to default. You can click on the drop down list and select 'Reset to Scaffold' option. Note that doing so would remove any code you have in the workspace that has not been submitted and you might want to save a backup before doing so.

ONLINE EXAM INSTRUCTIONS

- All submissions will be processed by similarity checks on Ed and submissions with high similarity score will be submitted as misconduct cases. **All submissions should be originally from yourself.**
- Only the .c file specified in the question will be marked, please do not rename the given files and please remove any other files except the required files before submitting your answer.
- After the due time, the assessment and answers cannot be accessed again.
- You have 3 hours window to complete the final assessment.
- Subject coordinator and tutors will be contactable by MS teams during the 3 hour window.
- Please do NOT post any questions in Canvas, Ed or Microsoft Team about the questions in the final exam.

ONLINE EXAM INSTRUCTIONS

- Demo questions for final assessment will be available on Ed. The demo questions are only for final assessment environment practice, the actual content of the final assessment questions will be different.
- Please test the demo questions for your access to Ed and submissions on Ed before the final assessment to make sure that you have the correct setup on your own device and browser.

ONLINE EXAM INSTRUCTIONS

- There will be multiple questions in the assessment.
- For the questions need you to explain concepts, please write your answer in dedicated place (e.g. comment area) in the program.
- For the questions need you to write a function, a part of a program or a whole program, you need to make sure that:
 1. Your submitted answer(program) can be successfully compiled on Ed without any errors or warnings using `gcc -Wall -Werror -ansi....` Command in the terminal. If your submitted answer (program) cannot pass the compilation, you will receive 0 mark for the question.
 2. Your submitted answer(program) attempts to complete the requirements in the question description. If your submitted answer(program) can compile but it does not attempt to complete the requirements, you will receive 0 mark for the question.
 3. Your submitted answer(program) is not hard-coded. Any hard-coded program will receive 0 mark for the question.