**Computer Programming (C20013) Examination**

### Date: 09/04/2013 - Allowed Time: 2 Hours

Procedure:

There are 4 questions. Answer only 3 questions. Answer all parts of these 3 questions.

Each question carries 40 marks. There are a total of 120 marks available.

***100% from this exam will contribute 40% towards your final grade.***

***Note:*** The use of any kind of electronic equipment is not permitted during the exam. Write your answers on A4 Paper (plain or with lines). Ensure all of your work is submitted to the teacher at the end of the examination. Write your name on every A4 page you use. Examination attendance must be signed to receive a grade.

### Question 1 (40 marks)

1. List three data types in C and describe when it is appropriate to use each.

Three data types in C are int, float and char. An int should be used to store whole numbers. A float should be used to store real numbers. A char should be used to store an individual character.

1. Explain the syntax and semantics of an “if” statement in C.

Semantics: an if statement is a sequence of statements that should only be executed if a particular boolean expression is true.

Syntax: The keyword “if” is followed by parentheses that enclose a boolean expression. The parentheses are follow by a pair of curly braces that enclose a sequence of statements.

e.g. if (my\_bool == true){my\_bool = false;}

1. Explain the syntax and semantics of an assignment “=” statement.

Semantics: An assignment statement gives a variable a specified value.

Syntax: A variable must be on the on the left hand side of an equals (=) sign. A variable or value must be on the right hand side of the equals sign. Like all other statements in C, an assignment statement must end with a semi-colon.

e.g. my\_int = 5;

1. Write C code that adds two hard-coded numbers together. Print out “Sum is greater than 10” if the sum is greater than 10.

int sum = 2 + 3;

if (sum > 10) {

printf("Sum is greater than 10.");

}

1. Write C code that takes an integer grade from the user. The program should print “Distinction” for a grade from 80 to 100, “Merit” for a grade from 65 to 79, “Pass” for a grade from 50 to 64, and “Invalid grade” for all other grades. This should be done with the use of an “if” statement, with “if else”, and “else”.

int main() {

int grade;

printf("Enter grade: ");

scanf("%d", &grade);

if (grade >= 80 && grade <= 100) {

printf("Distinction");

} else if (grade >= 65 && grade <= 79) {

printf("Merit");

} else if (grade >= 50 && grade <= 64) {

printf("Pass");

} else if (grade >= 0 && grade <= 100) {

printf("Fail");

} else {

printf("Invalid grade");

}

return 0;

}

1. List and describe three logical operators in C. Provide an example of how each logical operator works.

The logical AND in C, represented as &&, requires two boolean expressions as input and will return an output of TRUE if both expressions are true, or returns FALSE for any other permutation of its input. e.g. true && true == true; true && false == false;

The logical OR in C, represented as ||, requires two boolean expressions as input and will return an output of TRUE if either expression is true, or returns FALSE if both expressions are false. e.g. true || false == true; false || false == false;

The logical NOT in C, represented as !, requires one boolean expression. The value of the expression following the ! is inverted. e.g. !true == false; !false == true;

1. What is the difference between a compiler and an interpreter?

A compiler converts a high-level programming language code into efficient low level code with added optimizations. The compiled code is executed when running the program. An interpreter has no compilation stage for created more efficient code. The code that is written is exactly what will be executed. The interpreted code is executed on-the-fly.

1. Write C code that uses a nested for loop to draw the graphical shape of a right-angled triangle of asterisks like so:



int row, col;

for (row = 1; row <= 5; ++row) {

for (col = 1; col <= row; ++col) {

printf("\*");

}

printf("\n");

}

### Question 2 (40 marks)

1. What is a program?

A program is a sequence of instructions that a computer can interpret and execute.

1. What is a programming language?

A programming language is a language designed to facilitate the writing of computer programs.

1. Describe two different generations of programming languages with languages. Include a language that belong to each generation in your answer.

Third generation languages are high-level languages and as such, don’t require the programmer to know the address of each variable. Third generation languages introduced the use of structures, named variables and functions. C is a third generation programming language.

Fifth generation languages are high-level languages that model problems as a set of constraints, data and logical operators. These constraints, data and operators are used to solve a problem without being told by a programmer how to solve it. Prolog is a fifth generation programming language.

1. What are the relative advantages and disadvantages of each of the two language generations you described in the previous question?

Third generation languages can give more power the programmers by allowing them to specify exactly how a program should be executed. Third generation languages problems can be more intuitive to program than fifth.

Fifth generation languages offer a higher level of abstraction than third generation languages. Fifth generation languages can easily dynamically infer information.

1. List the stages in the construction a loop.

A loop termination variable that is referred to in the guard condition of the loop is given a value. The guard condition of the loop is evaluated. If the guard condition is satisfied, the statements inside the loop execute. The loop termination variable is altered and the guard condition is evaluated again. The loop continues until the guard condition evaluates to false.

1. Explain the syntax and semantics of a “while” statement

Semantics: A while loop is a logical construct the continues to execute the instructions inside it while a particular boolean expression belonging to the construct evaluates to true.

Syntax: The “while” keyword is followed by parentheses enclosing a boolean expression, followed by a pair of curly braces enclosing statements.

e.g. while (my\_int < 10){ ++my\_int; }

1. Explain the syntax and semantics of a “for” loop in C.

Semantics: A for loop is a logical construct the continues to execute the instructions inside it while a particular boolean expression belonging to the construct evaluates to true. The for loop has an initialization, evaluation and alteration stage that is usually performed on a loop termination variable.

Syntax: The “for” keyword is followed by parentheses enclosing three expressions separated by semi-colons. The first expression initializes a variable. The second is a boolean expression that determines if the loop should continue. The third expression usually alters the loop termination variable. The parentheses are followed by a pair of curly braces enclosing statements.

e.g. for (i = 0; i < 10; ++i){ printf(“i == %d”, i); }

1. What is an array?

An array is a data structure that can hold a number of variables of the same type. It is useful when the programmer doesn’t want to explicitly create a huge number of variables of the same data type.

1. With the use of an example, explain why is the array data structure is necessary?

An array data structure is necessary because there may be a situation where a programmer requires many, many variables of the same type. It would take a prohibitively long time to declare and assign these variables. Consider a situation where a program must store 1000 ages as integers and initialize them all to 0. This might be written:

int age1 = 0;

int age2 = 0;

…

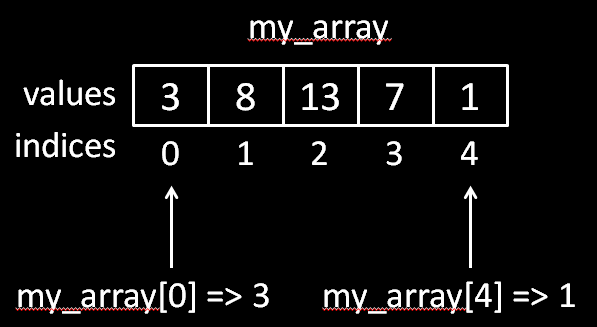
int age1000 = 0;

This is easily be shortened to:

int ages[1000] = {0};

1. What is the difference between the value of an array element and its corresponding index? Include a relevant diagram.

The index of an array element is the position of an element in the array relative to the beginning of the array. The value of an array element is the value stored in memory at the location of the array index.



1. Write C code that populates an integer array with 5 numbers (hard-coded) and searches the array for a certain (hard-coded) number.

int nums[5] = {2, 5, 31, 9, 4};

int i;

for (i = 0; i < 5; ++i) {

if (nums[i] == 31) {

printf("Found 31!");

}

}

1. Write C code that populates an integer array with 5 numbers (hard-coded) and print out the sum of the values in the array using:
   1. a “while” loop.

int nums[5] = {2, 5, 31, 9, 4};

int sum = 0;

int i = 0;

while (i < 5) {

sum += nums[i];

++i;

}

printf("sum: %d", sum);

* 1. a “for” loop.

int nums[5] = {2, 5, 31, 9, 4};

int sum = 0;

int i;

for (i = 0; i < 5; ++i) {

sum += nums[i];

}

printf("sum: %d", sum);

### Question 3 (40 marks)

1. What is the difference between system software and application software? Include three examples of each.

System software are files and programs that make up an operating system. They interact with the hardware and provide a platform for application software. Example of system software are graphical user interface libraries, sound libraries and operating system utilities.

Application software are built on top of system software. Examples of application software are Microsoft Word, accounting software and media players.

1. What is a sentinel? Include C code with a while loop and a sentinel to better explain a sentinel.

A sentinel is a special value used to guarantee the termination of a loop. It is useful when the number of inputs isn’t known in advance. In the following example a sentinel is set to -1 so that user input numbers will keep being added to the sum until the user enters -1.

int user\_input = 0;

int sum = 0;

int sentinel\_value = -1;

printf("Please enter a positive number to be added to the sum" );

printf(" or -1 to quit: ");

while (user\_input != sentinel\_value) {

sum += user\_input;

scanf("%d", &user\_input);

}

printf("\nSum: %d", sum);

1. What is top-down development?

Starting with a concept, the concept is continually decomposed in less abstract, more refined concepts until a level is reached where the concept no longer needs refinement. At this point, in computer programming, it is probably a single function.

1. Use the top-down development technique to describe an approach to programming an ATM.

ATM

* UI
  + keyboard
  + card reader
  + monitor
* …
* logic
  + check balance
  + withdraw cash
  + …
  + deposit cash
    - output cash
      * cash counter

1. List three uses of an editor for software development?

Three uses of an editor for software development are:

1. code completion
2. quick code refactoring
3. integrated debugging
4. Explain how to perform the following editor operations:
   1. Find and replace

Go to the window in the editor being used for find and replace, usually CTRL+R or CTRL+H. Type in the text to be replaced and the text to replace it with. Select a scope for the operation e.g. just one file, and hit “replace”.

* 1. Block copy

Highlight a block of text. Copy the text by pressing CTRL+C or via the “edit” option in the toolbar.

* 1. Block insertion

Having previously copied a block of text, press CTRL+V or “paste” via the “edit” option in the toolbar.

* 1. Block deletion

Highlight a block of text. Press “delete” or the back space arrow on the keyboard.

1. Explain, with examples, the following programming constructs:
   1. Input/Ouput

Input is data sent into a program e.g. which keys a user has pressed, text from a file, audio from a microphone. Output is data sent out from a program e.g. the result of a calculation, an image on the screen.

* 1. Cursor and screen handling

Cursor handling is the performance of operations on the cursor, e.g. move to a place requested by the player or changing icon depending on context (an egg timer when loading). Screen handling is the performance of operations on the screen, e.g. clearing the screen, updating windows, and setting the resolution.

1. Explain what a variable is

A variable is something that can change or be adapted. In C, a variable is a symbol associated with a data type and a memory address that holds a value.

e.g. int i = 0;

Here the data type is “int”, the symbol representing the variable is “i” and it is being given a value of 0, which will be stored in memory somewhere.

1. What does ASCII stand for?

American Standard Code for Information Interchange.

1. What is the ASCII table?

The ASCII table contains a standard character set for representing characters from the English alphabet. It consists of 128 characters. Each character consists of 1 byte. 7 bits of the byte are used to represent the character and the final bit is used for parity checking.

1. List the relational operators for the character data type in C and describe how the ASCII table is relevant to the relational operators.

The relational operators for the character data type in C are <, <=, ==, >= and >. The ASCII table is relevant to these relational operators because characters are compared based on their index in the ASCII table. e.g. ‘A’ < ‘a’ because ‘a’ comes later in the ASCII table and is therefore at a higher index.

1. What are control characters?

Control characters are characters that are reserved for special behaviours. In C, control characters take the form of an ASCII character proceeded by a backslash.

1. List three control characters and describe what each does.

Three control characters in C:

* \n – adds a line feed to text
* \t – adds a tab to text
* \0 – represents NULL and is important for use in strings of characters

1. Justify the statement: the ASCII table is an ordinal set of values.

An ordinal set of values is an ordered set. For examples, the months of the year have an order where January always come after December and before Febuary. The ASCII table is an ordinal set because they have a fixed order, i.e. first operations, then some symbols, then numbers, then letters.

1. What is the role of the extended ASCII set?

The extended ASCII set is a character set that inherits all characters from the ASCII table but uses the last bit (used for parity in ASCII) as to add 128 more characters to the table. These characters are used to add support for languages that use characters not present in the English alphabet.

1. What is a string? How is a string represented in C?

A string is a sequence of characters. A string in C is a array with the “char” data type.

e.g. char my\_string[] = “Hello, World!”;

1. Write C code to create and print a string.

char my\_string = "Hello, World!";

printf("%s", my\_string);

### Question 4 (40 marks)

1. Define a procedure.

A procedure is an ordered sequence of statements that does not return a value. Procedures can be used to reduce code duplication, enhance code readability and increase code reusability.

1. Explain the need for procedures.

Procedures are needed in large code bases to avoid code duplication, which is a common source of bugs. Procedures also allow code to be broken up into smaller chunks of code that can make code easier to read and understand. Rather than a programmer needing to read through several lines of code to understand what they do, he could know what they do from a well-named procedure.

1. Write down the standard syntax for a procedure definition in C.

The syntax for a procedure in C is as follows. The keyword “void” is followed by the procedure name, followed by parentheses enclosing procedure parameters. Each parameter is specified as a data type followed by a name. Parameters are separated by commas. A pair of curly braces follow the parentheses. Statements inside the curly braces are executed when the procedure is called.

1. Write a C procedure without parameters to print “Hello”.

void hello() {

printf("Hello");

}

1. What does it mean if a variable has function scope?

A variable with function scope can only be referenced inside its residing function. Any attempt to reference a variable with function scope outside of its residing function so will lead to a compiler error in C. Variables with function scope are de-allocated when the function is removed from the stack.

1. What does it mean if a variable has block scope? Give an example of a variable with block scope.

A variable with block scope can only be referenced inside its residing block. Any attempt to reference a variable with block scope outside of its residing block so will lead to a compiler error in C. Variables with block scope are de-allocated when the block has finished executing. Examples of blocks are for and while loops.

int i = 0;

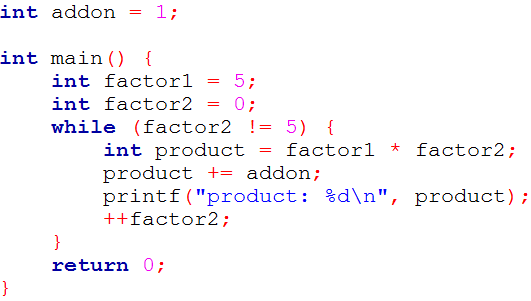
while (i < 10) {

int x = i \* 2; // x has block scope

++i;

}

1. Identify the scope of each variable in the following program:



addon: global scope.

factor1: function scope

factor2: function scope

product: block scope

1. What is a function?

A function is an ordered sequence of statements that returns a value. Procedures can be used to reduce code duplication, enhance code readability, increase code reusability and return the result of calculations performed elsewhere in the code.

1. What is the difference between user defined functions and standard library functions?

A user defined function is a function that has been created by a programmer for his particular work and can be very specific to that work. For example, a user defined function could find all customers with more than $10,000 in their account. Functions in the standard library have been created by many programmers over the years and are not particular to any problem. For example, concatenate one string to another.

1. Explain the difference between a function and a procedure.

A procedure has a void return type, a function can return any data type.

1. Define the length of a string.

The length of a string is the number of characters in that string. The length of a C string can be calculated by counting the number of characters that occur before the null character.

1. What is the difference between the length of a string and its dimension?

The length of a string is the number of characters in the string. The dimension of a string is the number of indices required to reference a character in the string.

1. Write C code to that creates a string a uses a “for” loop to calculate the length of the string.

char my\_string[] = "hello";

int length;

for (length = 0; my\_string[length] != '\0'; ++length) {}

1. Why is data validation needed?

Data validation is needed to ensure that data, particularly user input data, is of a format that is valid for further processing. Consider the example of a program could request that the user inputs a number but instead the user inputs a letter. Without data validation, the program would continue with this invalid data and crash soon after. However, data validation could check the input data to ensure that a number has been entered, giving an error and request for further input if a letter was entered.

1. What do the following boolean expressions evaluate to:
   1. true AND NOT true

false

* 1. (true AND false) OR true

true

* 1. (false OR NOT false) AND true

true