First Creation (Date - Version No.):

* 20070731-00 (draft from Japanese Expert)

Course Title: C Programming Introduction
Course Code: 02-06
Product Code: B2-03

Revision History (Date - Version No.)			
1		16	
2		17	
3		18	
4		19	
5		20	
6		21	
7		22	
8		23	
9		24	
10		25	
11		26	
12		27	
13		28	
14		29	
15		30	

Final Version (Date - Version No.) ：

Official Approval	Date of report to PIU

Semester:2 Course Title: C Programming Introduction Week: 1st:

Theme: Programming environment introduction, compiler, code editor

Japanese Expert			HUT Staff: Cao Tuan Dung	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Guidance the class		Introduce briefly about the course, its goal and its main topics and the rules need to be respected too.	3%
2	Environment and tools		List all the topic of this lesson. Operating System: Linux Fedora 8. Compiler: gcc Editor : emacs Window manager: KDE / Gnome	3%
3	Login		- All students have their account and password (provided by Hedspi). Login to system.	4%
4	Linux file system		Present the tree like structure of Linux file system, root directory and others important directories as /root, /dev /etc /bin and home directory	5%
5	Basic commands		Open a command window in Linux. Practice with the basic commands: cd, ls, cat, which, cp, man. Working with multiple command windows	10%
6	Exercise 1.1		Set up working directory for this course. Ask student to organize their home directory and create a sub directory to manage source codes, executable programs for each week.	10%
4	Steps to make a program		Explain steps in running a program, from using Emacs to editing to compile by gcc and run the executable output.	5%
5	Using Emacs		Introduce Emacs editor. Basic functionalities and commands. Open, save file, move around, copy,	20%

			paste	
6	Exercise 1.2 Emacs configuration		Setup font, colour, indentation for Emacs editor. Ask students to use Emacs to edit the configuration file of its self.	10%
7	Exercise 1.3		Edit the first C program using Emacs. Save it in the working directory. Practice and view the effect of Emacs configuration.	15%
8	Exercise 1.4, 1.5		Design simple algorithms for simple problems.	15%

Semester:2 Course Title: C Programming Introduction Week: 2nd
Theme: Introduction to C programming language

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	C Program Structure		Explain the general format of a c program. Give a very simple example and explain. explain the meaning of pre-processing as #include <stdio.h> To declare using the standard I/O library. Other libraries: string, time, math… An C program must declare only one main() function. The first line in the main() will implement when the program starts. The syntax to open and close a block of codes. The printf() function sends the output to standard output (monitor). This function will be taught in the next week. The goal of return statement is to Stop and Quit the program.	10%
2	Keywords of C		Students recognize some C language keywords, and know to not use them as identifiers.	5%

3	gcc compiler, syntax		Basic examples of using gcc to compile a program: • without specifying the name of output • indicating a name for output Compile to object file and Link files. Compile with a library	10%
4	Exercise 2.1		Compile the first program in C (result of previous exercise). Explain the default output: a.out Run program compiled and view the result.	10%
5	Dealing with error.		Students must understand what errors are, where they are in the source code and how to fix it.	10%
6	Exercise 2.2, 2.3, 2.4		Use gcc to compile the file hello.c in previous exercise to an executable program named sayhello and run it. Modify the source code to view the effect of end of line character in the function printf.	20%
7	Exercise 2.5		Write a program to output a greeting sentence with the introduction about the student, and his class.	10%
8	Exercise 2.6		Edit a predefined program and save it as pi.c. Try to understand the meaning of the code. Compile and run it. Review the code to understand the output.	10%
9	Exercise 2.7		Write a program that writes a program that writes the name of the person sitting next to you. Solution and explain briefly about special escape character	10%
10	Exercise 2.8		Write an algorithm: creating a simple C program	5%

Semester:2 Course Title: C Programming Introduction Week: 3rd
Theme: Standard output introduction

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Input/output in C		C has no built-in statements for input or output. A library of functions is supplied to perform these operations. The I/O library functions are listed the “header” file <stdio.h>.	2%
2	Formatting Output with printf		printf(format-control-string, other-arguments); - format control string: includes a listing of the data types of the variables to be output and, optionally, some text and control character(s). - other-arguments: correspond to each conversion specification in format-control-string	2%
3	Printing Integers, Floating-Point Numbers		Explain the conversion specifiers for intergers (d,i,u,x,...) and float-point numbers (f,g,e). Give an example for each type.	2%
4	Printing Strings and Characters		Char: %c. Cannot be used to print the first character of a string String %s: Requires a pointer to char as an argument Prints characters until NULL ('\0') encountered Cannot print a char argument. Give an example.	2%
5	Other Printing Conversion Specifiers		Explain conversion specifiers for displaying pointer value (%p), number of characters already output by current printf statement (%n) and Prints a percent sign (%%). Give an example.	1%
6	Printing with Field Widths and Precisions		Explain printing with field witdths precisions. Give an example.	2%

7	Printing Literals and Escape Sequences		Explain the literals and escape sequences. Most characters can be printed. Certain "problem" characters, such as the quotation mark ", must be represented by escape sequences. Represented by a backslash \ followed by an escape character.	1%
8	Exercise 3.1		Write a program that shows the size of basic data types, such as: int, long short, double, char… Give the solution for exercise 3.1 and explaination. Example: printf("short int %d\n",sizeof(short int));	8%
9	Exercise 3.2		Compile a given program and see what are display. Explain why.	5%

10	Exercise 3.3		Write a program that asks your name and then greets you.
			You can use scanf() function to read data with specified format from keyboard.
			E.g:
			char word[20];
			scanf("%19s", word);
			Solution:
			scanf("%15s", name);
			printf("Hi there, %s!\n", name);
				20%

11	Exercise 3.4		Write a C program that will read in two integers n and m and print out the sum of all the values between n and m inclusive Solution for exercise 3.4 and explanation	25%
12	Exercise 3.5		Write a program that compute the total cost of	30%

			DVD rental. Help student to understand the Input/Output of floating number, character.	

Semester: 2 Course Title: C Programming Introduction Week: 4th

Variables, constant, Standard input

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Identifiers, Variable		Repeat identifiers. Up to 31 chars (letters, numbers, including _), must begin with a letter.... Review basic knowledge about variable in language C. The basic concepts about variable: type, scope, lifetime.	3%
2	Basic data types, Variable declarations, definitions, initialization		Review basic knowledges about basic datat types in language C. The basic data types: char, int, double,... Give examples. Explain the use of keywords: definitions, initialization, extern variable and the usage of variable. Give an example about variable. e.g: printf(“%d + %d = %d\n“, a, b, c);	3%
3	Variables and Constants		Explain the difference between variables and constants. Variables are used to store values and we can change these values while constant's value is invariable during the program. Give examples about variables and constants.	2%
4	Formatting Input with Scanf		Format: scanf(format-control-string, other-arguments); -format-control-string - describes formats of inputs -other-arguments - pointers to variables where	2%

			input will be stored -can include field widths to read a specific number of characters from the stream. Give examples for scanf.	
5	Exercise 4.1		Write a program that reads a integer and a double from user, use a floating-point and an integer variable to store and then show to screen. Solution for exercise 4.1 and explanation	10%
6	Exercise 4.2		Write and run this program to see the limit of basic data types: int, long. Widen this program for other basic data types. Use limits.h library to build your programs. Solution for exercise 4.2 and explanation	15%
7	Exercise 4.3		Write a program that reads a string from the keyboard by using a scan set.	10%
8	Exercise 4.4		Write a program that inputs data with a field width. Widen to all basic data types.	10%
9	Exercise 4.5		Write a program take from users value for a radius, then output the area and the circumference of the circle corresponding. Finally, calculate the area and the volume of the sphere corresponding Note: Show the benefit of using constant	15%
10	Exercise 4.6		Write a program calculate the employee's wage for a week. Note: Ask students to change the format of output with various numbers after the point decimal.	15%

11	Exercise 4.7		Write a program display the bill that users have pay for buying a book from BK booksellers.	15%

Semester: 2 Course Title: C Programming Introduction Week: 5th

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Arithmetic Operators Equality and Relational Operators		Briefly recall these basic operators in C by giving examples Less than < a < 5 Less than or equal <= a <= b More than > a > b + c More than or equal>= a >= b + 5 Equal == a == -6 Not equal != a != 0	2%
2	Logical Operators Bitwise Operators		AND && (a > 0) && (b > 0) OR || (a <= 0) || (b <= 0) Negation ! !(a && c)	2%
3	Assignment Operators and Expressions., Increment and Decrement Operators		Explain the different of prefix and postfix increment (decrement) operators.	3%
4	Conditional Expressions		expr1 ? expr2 : expr3 If expr1 is true do expr2 If expr1 is false do expr3 Give an example.	3%
5	Exercise 5.1		Write a program that converts distances from kilometres to miles. Ask user to input the kilometres value then	3%

			output to screen the miles value.	

6			scanf("%lf", & kms);
			
			/* Convert the distance to miles. */
			miles = 1000 * kms;
				7%

7	Exercise 5.2		Run the exercise5_2.c program below to illustrate the operation of Logical operators and relational operators.
			Replace b - a == b – c by a = b-c and then Explain the result.
			#include <stdio.h>
			main()
			{
			int a = 5, b = 6, c = 7;
			puts("int a = 5, b = 6, c = 7;\n");
			printf("The value of a > b is \t%i\n\n", a > b);
			printf("The value of b < c is \t%i\n\n", b < c);
			printf("The value of a + b >= c is \t%i\n\n", a + b >= c);
			printf("The value of a - b <= b-c is\t%i\n\n", a - b <=b- c);
			printf("The value of b - a == b – c is\t%i\n\n",b- a==b- c);
			printf("The value of a * b != c * c is\t%i\n\n",a * b<c * c);
			}
				15%

8	Exercise 5.3		Type and compile the exercise5_3.c below, the program illustrates the operation of conditional expressions. Alter the program by eliminating the abs and	20%

			max variables.
			main()
			{
			  int n, m, abs, max;
			  printf("Enter a positive or negative integer: ");
			  scanf("%i", &n);
			   printf("\nYou entered %i.\n", n);
			  abs = n < 0 ? -n : n;
			  printf("Its absolute value is %i.\n", abs);
			  printf("\nEnter two integers (e.g. 1 2): ");
			  scanf("%i %i", &n, &m);
			
			  printf("\nYou entered %i and %i.\n", n, m);
			  max = n > m ? n : m;
			  printf("%i is the larger value.\n", max);
			}	

9	Exercise 5.4		Illustrates the integer overflow that occurs when an arithmetic operation attempts to create a numeric value that is larger than can be represented. Type and compile the program to see the result.	15%
10	Exercise 5.5		Write a program that requires user to input two double values stored in two variables x,y. Use if control structure to examine all the relation between x and y.	15%
			Solution for exercise 5.5 and explaination	
11	Exercise 5.6		Write different expressions, each show out the result to a requirement of dividing a group of student into class.	15%

Semester: 2 Course Title: C Programming Introduction Week: 6th

Theme: Branches statement

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	The if Selection Structure		Explain the usage of the if selection structure and the syntax of if selection. Explain the basic concepts: decision symbol.	3%
2	The if/else Selection Structure		if Only performs an action if the condition is true. if/else A different action when condition is true than when condition is false. Explain the usage of Ternary conditional operator, Nested if/else structures, and Compound statement.	2%
3	The switch Multiple-Selection Structure		Useful when a variable or expression is tested for all the values it can assume and different actions are taken. Explain the format of Switch selection and give an example.	2%
4	Exercise 6.1		Write a program that finds and displays the alphabetically first letter in a sequence (e.g. type IBK and it returns B). Note the use of a normal if/else and then an if on its own. Solution and explanation.	10%
5	Exercise 6.2		Write a program that transforms a compass heading to a compass bearing using this table: 0 - 89.999... north (heading) east	13%

			90 - 179.999... south (180.0 - heading) east 180 - 269.999... south (heading - 180.0) west 270 - 360 north (360.0 - heading) west Solution and explanation. Show the elegant use of if and else if structure.	
6	Exercise 6.3		Write a program that requires you enter an age and shows you what your class is. (child, Senior Citizen or adult) Child : age <18 Adult : 18<=age<65 Senior Citizen: age >=65 This program should uses the if/else structure as a building block in a more complicated structure. It works out a category for people based on their age. Note the way that the if/else structure actually contains another if/else structure. Solution and explanation	10%
7	Exercise 6.4		Modify exercise6.3 by using if/else structure with conditionals. This is a variation on the age program above; see if you can understand the use of conditionals used. You must always be careful about ranges when using the conditionals. What happens if you enter 18 as your age and how can we solve this problem?	10%
8	Exercise 6.5		Write a program to play “High/Low". The program “picks" a number. The human player tries to guess it. The program indicates if the	15%

			guess is too high, too low, or correct. Then it stops. Solution and explanation: Use rand() function to pick a random number.	
9	Exercise 6.6		Write a program that reads in three integers and then determine which one is the smallest, and display it. If the values are a, b, and c, there are four cases: a is smallest if a < b and a < c b is smallest if b < a and b < c c is smallest if c < a and c < b Number is smallest When?	10%
10	Exercise 6.7		Alter the exercise 6.3 by using Switch selection structure. Alter the exercise 6.5 by using Switch selection structure.	15%
11	Exercise 6.8		Write a program calculate the parking fees for various kinds of vehicle. Solution: Using else if statement	10%

Semester:2 Course Title: C Programming Introduction Week: 7th

Theme: Loops (1)

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	The for Repetition Structure		Explain the format of For structure that using for loops: for (initialization; loopContinuationTest; increment) statement Explain the details about: Initialization, loop-continuation, and increment. Give examples.	2%

2	Notes and Observations		Some notes about Arithmetic expressions, loop continuation condition and Control variable. Give an example.	3%
3	Exercise 7.1		Write a program that prints ten integers and their squares. 1 1 2 4 3 9 ... 10 100	5%

4			Solution:
			for(i = 1; i <= 10; i = i + 1)
			printf("%d %d\n", i, i * i);	

5	Exercise 7.2		Write a program that prints out a triangle like: * ** *** **** ***** ******	10%

6			Just to show the case to use nested for loop
			for(i = 1; i <= 10; i = i + 1)
			{
			for(j = 1; j <= i; j = j + 1)
			printf("*");
			printf("\n");
			}	

7	Exercise 7.3		Write a program that lists numbers which is greater than 27 from 1 to 100. Solution: Combine for and if statement	7%

8	Exercise 7.4		Write a program that lists prime numbers which is smaller than 100. Use math.h library to use some mathematical functions: sqrt,... Solution for exercise 7.4 and explanation	13%
9	Exercise 7.5		Alter the exercise 7.4 above by eliminating the even numbers to avoid calling sqrt function many times.	10%

10	Exercise 7.6		Show the effect of a loop in a reverse direction. Try the following program in your compiler.
			#include <stdio.h>
			int main(void)
			{
			int time, start;
			printf("Enter starting time (an integer) in seconds> ");
			scanf("%d", &start);
			printf("\nBegin countdown\n");
			for (time = start; time > 0; time = time - 1)
			{
			printf("T - %d\n", time);
			}
			printf("Blast-off!\n");
			return (0);
			}
				5%

11	Exercise 7.7		Write a program that converts temperatures from Celsius to Fahrenheit. Notice the conditions of the loops continuation and the way in which #define macros are used to	15%

			set constant values. fahrenheit = 1.8 * celsius + 32.0 Solution for exercise 7.7 and explanation	
12	Exercise 7.8		Sometimes we need to have loops within loops, this is called nested loops. This program demonstrates how this works. By running it you should see in what sequence the code is called. Solution for exercise 7.8 and explanation	10%
13	Exercise 7.9		Write a program that uses for structure to calculate the value of n!.	10%
14	Exercise 7.10		In mathematics, a perfect number is defined as a positive integer which is the sum of its proper positive divisors, that is, the sum of the positive divisors not including the number itself. E.g: 6=1+2+3 Write a program that lists perfect numbers which is smaller than inputted N.	10%
15			Explain how to solve exercise 7.10	

Semester: 2 Course Title: C Programming Introduction Week: 8th
Theme: Loops (2)

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.

1	while, do while statement		Rapidly review how these statement works.
			while (expression) {
			Statement1;
			Statement2;
			...
			}
			And give examples.
				6%

2	Break and Continue Statement		The break statement provides an early exit from for, while, and do while loop. The continue statement is related to break, but less often used; it causes the next iteration of the enclosing for, while, or do loop to begin.	4%
3	Exercise 8.1		Write a program that copies content inputted from the keyboard to the screen, but replace the sequence of blank characters by only one blank character. You can use getchar() and putchar() method to carry out this program. Solution for exercise 8.1 and explanation	15%
4	Exercise 8.2		Write a program that replaces characters such as: tab, \t,\b by \\ character in the input string and print out. You can use getchar() method to carry out this program. You can use if structure or switch structure.	15%
5	Exercise 8.3		Calculate square cube by using Newton method. Focus on the use of while or do while loop in this situation when programmer do not know in advance the number of loops.	15%
6	Exercise 8.4		Write a program that use while structure to analysis of examination results: how many passed students and failed students. You can simply ask user to show that a student is passed or failed by entering a presented number: 1 is passed and 2 is failed. Explain the solution.	7%

7	Exercise 8.5		Use do...while statement to print out integers that is smaller than a preceded number. Note that the do...while statement always performs one time at least.	8%
8	Exercise 8.6		We would like a program to average a set of grades. Algorithm notes: We need a running sum of grades, and a running count of how many grades have been read so far. We need to read until we get a sentinel value | let's use a negative grade to indicate we are done. Need to be sure we print prompts.	15%
9	Exercise 8.7		Write a program that compute n! using a loop. You can use: Counter" variable, i, ranging from 1 to n. Running product f, tracking i!.	15%

Semester: 2 Course Title: C Programming Introduction Week: 9th

Theme: Functions

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Concept of function in C		What is a function in C? A subprogram that we can call for many time. a group of declarations and statements that is assigned a name, usually has a value	1%
2	Function prototype, function call		Explain the example: function square(). How it is defined outside the function main(), how it is called by main()	2%
3	Procedural programming		The goal of functions. Break your problem down into smaller sub-tasks and solve more easily complex problems. Function help to generalize a	1%

			repeated set of instructions too.	

4	Characteristics of Functions, arguments, return.		When invoking a function call, we can include function parameters in the parameter list. Except void function, all functions return a value.
			return-type name(argument-list)
			{ local-declarations
			statements
			return return-value;
			}
				2%

5	Exercise 9.1		Write a function to calculate the kinetic energy of the element ke= $mv^2/2$, for m is mass (kg) and v is speed (m/s) Use this function in a program.	5%
6			Solution for exercise 9.1 and explanation: the function should take two parameters: m and v	3%
7	Exercise 9.2		Write a function is_prime that accepts a positive integer and returns 1 if it's a prime number and 0 otherwise. prototype: int is_prime(int n); Now write a program that gets a positive integer from the user and prints all the prime numbers from 2 up to that integer. Use the function is_prime(n)	10%
8			Solution for exercise 9.2 and explanation: a number n is prime if it is not the multiple of any number from 2 to square root of n.	4%
9	Exercise 9.3		Write programs to setup these following functions. Use them in a main program - A function to find the sum of the cube of integers from 1 to n - A function to list all submultiples of the integer n	8%

			- A function to list the n first perfect square numbers	
10			Solution for exercise 9.3 and explaination: one parameter n and use the for loop	5%
11	Exercise 9.4		Write a program to calculate the worker's salary by a week. The average wage is 15000 VND for one hour working. And workers have to do 40 hours a week. If they work overtime, the money is paid more 1.5 times for each hour.	10%

12			Solution for exercise 9.4: write a function salary take the number of working hours as parameter.
			if (hours >40)
			return 15000*40+15000(hours-40)*3/2;
			else return hours*40;
				5%

13	Exercise 9.5		Write the function: void printnchars(int ch, int n) to display a character for n time. Use this function to print "* - triangle" which has edges of 4, 5.	4%
14			Solution for exercise 9.5 and explanation	3%
15	Exercise 9.6		The formula for converting a temperature from Fahrenheit to Celcius is C = 5/9(F-32) Write a function named celsius that accepts a Fahrenheit temperature as an argument. Function should return the temperature in Celcius. Display a table of the Fahrenheit temperature 0 though 20 and their Celsius equivalents.	7%
16			Solution for exercise 9.6 and explanation: Use a while loop to draw a table of temperature. Step 1 degree for each line in table	4%
17	Exercise 9.7		Given a positive number n, this is a k-figure	7%

			number. Write a function to verify whether n has all figures being odd numbers or even numbers.	
18			Solution for exercise 92.7 and explanation: Use two flag variables: even and odd and switch them according to each digit in the number. To take all the digits in a number positive: use modulo of 10.	6%
19	Exercise 9.8		The program Vietnamese Idol has 5 judges, each of whom awards a score between 0 and 10 for each performer. Performer's final score is determined by dropping the highest and lowest score received the averaging the 3 remaining scores. Write a program that uses this method to calculate a contestant's score using two following functions: void getJudgeData() should ask the user for a judge's score, store it in a reference parameter variable, and validate it. void calcScore() should calculate and display the average score of performer.	8%
20			Solution for exercise 9.8 and explanation. calcScore will take 5 parameters. You need to implement max and min function.	5%

Semester: 2 Course Title: C Programming Introduction Week: 10th
Theme: Arrays

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Array data type		Explain what an array is and give some examples. - list of students' marks	1%

			- series of numbers entered by user - vectors - matrices	
2	Array in the memory		Sequence of variables of specified type The array variable itself holds the address in memory of beginning of sequence. Show the figure of array's elements in the memory. The n-th element of array arr is specified by arr[n-1] (0-based)	2%
3	Array initialization		Give example and explain about how to initialize the array: - during declaration. - With the use of for loop in the case the number of elements is only known in runtime.	2%

4	Array input and output		Illustrate the use of for statement in a program that takes the input for the rainfalls of all the month of the year, then displays it in a table
			for (i=0; i < MONTHS; i++){
			printf("Enter the rainfall(mm):");
			scanf("%d", &rainfall[i]);
			}
			for (i=0; i < MONTHS; i++) {
			printf("%5d ” , rainfall[i]);
			}
				3%

5	Exercise 10.1		Write a program to input an array that stores 100 integers. a) Find the sum of the odd number in the array b) Find the minimum value.	3%
6			Solution for exercise 10.1. Combination of for	7%

			statement and if statement. Be careful with the index of elements, so that it does not exceed the limit.	
7	Exercise 10.2		Given an array of which elements are the numbers inputted by user. Find the sum of the local maximum numbers in this array (local maximum element is the element that greater than its two neighbours)	3%

8			Solution for exercise 10.2
			... // Array data input
			sum=0;
			for (i=1; i<=size-1; i++)
			    if (a[i]>=a[i-1] && a[i]>=a[i+1]) sum +=a[i];
				7%

9	Arrays as function arguments		example: int calc_sum(int arr[], int size); Within the function, arr is accessed in the usual way Changes in the function affect the original array!!	2%
10	Exercise 10.3		Implement a function that accepts two integer arrays and returns 1 if they are equal, 0 otherwise Write a program that accepts two arrays of integers from the user and checks for equality	4%
11			Solution for exercise 10.3: Use a *for* loop to compare all two element at the same index in two arrays.	6%
12	Exercise 10.4		Write two functions: The first sorts the integers in an array by the decreasing order. The second sort the odd elements in the decreasing order. Write a program that asks user to enter 10	4%

			integers and displays the results after two styles of sorting above.	

13			Solution for exercise 10.4 and explaination:
			for (i = 0; i < n-1 ; i++)
			for (j = i+1; j < n; j++)
			if (a[i]<a[j] && (a[i]%2) && (a[j]%2)) {
			tmp=a[i];
			a[i]=a[j];
			a[j]= tmp;
			}
			}
				11%

14	Exercise 10.5		Given an integer array: Count the number of the number 0 in this array. Find the length of the subsequence that consists of consecutive numbers (all of elements are number 0). Find the time of appearance of numbers.	5%
15			Solution for exercise 10.5: a) Traverse the array, when program meet an element 0, increment the counter. c) Firstly we sort the array, and then count the number of equal elements just after each element.	10%
16	Multi-dimensional arrays		Array of arrays: int A[2][3] = { {1, 2, 3}, {4, 5, 6} }; Access: j-th element of the i-array is A[i][j]. Explain the program do the addition of two matrices	1%
17	Exercise 10.6		Write a program that defines 3 matrices A,B,C of size 3x3 with int elements; initialize the first two matrices (A and B)	3%

			Compute the matrix multiplication of A and B and store it in C (i.e. C = A*B) Print all the matrices on the screen.	
18			Solution for exercise 10.6 and explanation. Solution is similar with what we do in the previous example.	6%
19	Exercise 10.7		Input an array with the number of element n asked from user. Check to see whether the array is symmetric.	3%
20			use two cursors i, j: i start the beginning and j from the end element of array. Compare a[i], a[j] then increment i, decrement j	7%
21	Exercise 10.8		Write a function that reverses the array content. Use this function in a program that asks user to enter a list of floating numbers. Then reverse all these numbers without creating another array.	3%

22			Solution for exercise 10.8. :
			for (i=0; i<n/2; i++)
			  tmp=a[i]; a[i]=a[n-i-1];
			  a[n-i-1]=tmp;
				7%

Semester: Course Title: C Programming Introduction Week: 11th
Theme: Pointers

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Memory address		Computer's memory is made up of bytes. Each byte has a number, an address, associated with it.	

			Show a figure for illustrating.	
2	Address operator		Introduce the use of & operator. Give an example of a program that prints out the value and the address of a variable	
3	Exercise 12.1		Write a C program to input three integers. Set up a single pointer to point to each of these integers in turn. Display the value dereferencing the pointer.	
4			Solution for exercise 12.1 and explanation.	
5	Exercise 12.2		Write a program that print out the address (in hexadecimal format) of first 5 elements of an array.	
6			Solution for exercise 12.2 and explanation.	
7	Pointer, declaration of pointer variable		Explain a pointer as a variable that hold the address of an memory cell. A pointer should have the same type as the variable or array that it points to.	
8	Referencing, dereferencing		To make a pointer to point a variable, you must use & operator. ptr = &c; To print a pointer, use %p format access indirect to the content of a variable via pointer using * operator	
9	Exercise 12.3		Write a program asking the value from user for 3 float variable a, b, c. Then add 100 to the content of them by using just a pointer.	
10			Solution for exercise 12.3	
11	Pass arguments by value		we saw until now received their arguments “by value”	

			They couldn't change values in the calling function. Show a example on two versions of swap function: without and with the pointer parameters.	
12	Exercise 12.4		Write a function that takes three variable (a, b, c) in as separate parameters and rotates the values stored so that value a goes to be, b, to c and c to a. Test this function in a program	
13			Solution: just use one temporary local variable in the swap function. All parameters must be pointers.	
14	Exercise 12.5		Introduce int variables x, y, z and int* pointer variables p, q, r. Set x, y, z to three distinct values. Set p, q, r to the addresses of x, y, z respectively. 1) Print with labels the values of x, y, z, p, q, r, *p, *q, *r. 2) Swapping values of x, y, z. Print with labels the values of x, y, z, p, q, r, *p, *q, *r. 3) Swapping values of p, q, r. Print with labels the values of x, y, z, p, q, r, *p, *q, *r. Give the solution.	
15	Exercise 12.6		To increase salary for an employee, write a function `incomeplus` that is based on the current salary and the number of years passed from the beginning years (must > 3) of current salary. Test it in a program.	
16			Solution: write a function to calculate salary using pointer as parameter.	

Semester: 2 Course Title: C Programming Introduction Week: 12th

Theme: Arrays and pointers

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Pointers and arrays		The name of an array refers to the address of the first element. When this name is assigned to a pointer of the same type with array, the pointer will point to array. int s[10]; int *iptr; iptr=s; /* From now iptr is equivalent to s */ Both iptr and s now point to s[0] Note: Unlike pointers, the value of an array name cannot be changed	3%
2	Pointers arithmetic		Pointers can be incremented and decremented If p is a pointer to a particular type, p+1 yields the correct address of the next variable of the same type	2%
3	Distance between pointers		If p and q point to elements in an array, q-p yields the number of elements between p and q. With the use of ++ or -- operator on a pointer, we can pass all array elements. Demo in a example.	1%
4	Passing array to function		Show another way to pass an array to a function using pointer. Just pass the pointer to the fisrt element.	2%
5	Exercise 12.1		Write a function countEven(int*, int) which receives an integer array and its size, and returns	3%

			the number of even numbers in the array.	

6			Solution for the exercise 12.1 and explaination
			int count =0;
			for (i=0; i<size; i++)
			  if (*(arr+i)%2==0) count++;
				9%

7	Exercise 12.2		Write a function that returns a pointer to the maximum value of an array of double's. If the array is empty, return NULL. double* maximum(double* a, int size);	5%
8			Solution and hint: Use a pointer p to traverse the array. A pointer max is initialized with the address of first element. Element pointed by Max is compared with elements pointed by p.	10%
9	Exercise 12.3		Write a function getSale uses a pointer to accept the address of an array. It asks the user to enter the sales figures and stores those figures in the array. Write a function totalSale return the total of the element int the array. Use these two functions in a program to input the sales figure from different quarteurs and display the total. Using pointers instead of array in function's parameters.	5%
10			Give the prototype of two functions: getSale and totalSales	15%
11	Exercise 12.4		Write a program to list all the sub array of an	3%

			given array.	
12			Give and explain the solution for this exercise. The most difficult thing is to find the right algorithm. Help student to find it out.	22%
13	Exercise 12.5		Write a program to reverse an array in two different ways: using indexes and using pointers. Solution for exercise 12.5 and explanation.	20%

Semester: 2 Course Title: C Programming Introduction Week: 13th
Theme: Strings

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Strings as array of characters		Recall about the concept of strings in C. Give somes examples Another way to initialize: char str[] = "Text";	2%
2	Null character		A string terminates with a '\0' character. Show an illustration in a figure of memory storage for a string. It means that in order to hold a string of N characters we need an array of length N + 1	2%

3	String manipulation functions		Explain the purpose of functions strcat, strncat, strcmp, strcpy. Give an example for the usage of strcat, strncat:
			char s1[20] = "Happy ";
			char s2[] = "New Year ";
			char s3[40] = "";
			printf("s1 = %s\ns2 = %s\n", s1, s2);
			printf("strcat(s1, s2) = %s\n", strcat(s1, s2));
				3%

4	Character analysis and conversion		Introduce functions in library ctype.h such as isupper(), islower, toupper(), tolower()	2%
5	Array of strings		a two-dimensional array of characters in which each row is one string. Show some examples	1%
6	Exercise 13.1		Write a program that inputs a line of text, counts the number of blanks by using a function, and displays the number of blanks	3%
7			Solution for exercise 13.1 and explanation. Write a function `spacecounter` for counting blanks. Just do a loop for to traverse all characters of strings.	12%
8	Exercise 13.2		Write a function that: gets a string and two chars the functions scans the string and replaces every occurrence of the first char with the second one. Write a program to test the above function the program should read a string from the user (no spaces) and two characters, then call the function with the input, and print the result. For example input: “papa”, ‘p’, ‘m’ output: “mama”	4%
9			Solution for exercise 13.2 and explanation. Develop a replace function using a loop verifing if a character in string is needed to replace.	6%
10	Exercise 13.3		Write a program that tests a customer number to	5%

			determine whether it is in the proper format (LLLNNNN with LLL are letters and NNNN are numbers).	
			Solution for exercise 13.3 and explanation. Use functions in library ctype.h	10%
11	Exercise 13.4		Write your own replacement for the standard strcpy() that comes with C without using string.h	5%

			Solution for exercise 13.3 and explanation.
			char *my_strcpy(char *destination, char *source)
			{
			    char *p = destination;
			    while (*source != '\0') *p++ = *source++;
			    }
			    *p = '\0';
			    return destination;
			  }
				10%

12	Exercise 13.5		Write a program asks the user to enter his or her first and last names, separated by a space. Then print out the first name.	5%
			The program shoud use a function `nameslice` which cuts off the last name off the string in parameter. It scans the array looking for a space. When it finds one, it replaces it with a null terminator.	10%
13	Exercise 13.6		Write the function strend(s,t), which returns 1 if the string t occurs at the end of the string s, and zero otherwise. Solution	10%
14	Exercise 13.7		A list of product number and description of shop is: "TV127 31 inch Television",	5%

			"CD057 CD Player", "TA877 Answering Machine", "CS409 Car Stereo", "PC655 Personal Computer" Store this list in an array of string and write a program allowing user to lookup a product description by entering all or part of its product number.	
			Solution for exercise 13.7 and explanation. Showing the usage of strstr() function.	10%

Semester:2 Course Title: C Programming Introduction Week: 14th
Theme: Structures

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1	Struct data type		Review basic knowledge about structure in language C. The purpose of usage of the struct type in program: Structures, or structs, are very useful in creating data structures large and complex.	2%

2	Normal declaration of struct and variables		Explain the use of keyword: struct and the specification of fields.
			struct student {
			  char name[20];
			  int age;
			  float grade;
			  };
			Note to students: Use struct before type name in declaration of variable
				4%

3	Structure type definition		typedef struct car {	4%

	with typedef		char* make;
			char* model;
			int year;
			}car_t;
			Note: With typedef, it's no need to use the keyword struct in the declaration of variables.	

4	Exercises 14.1		Create user defined for Date type. Write functions for manipulating data of this new type. a) Create a structure named Date for storing date concerning variables. Each date has a day, a month and a year. b) Write a function for the input of variable of this type. Remember to check the validation of data c) Write a function to datecmp to compare two date which return -1 if the first date (parameter) is before the second 0 if two date are identical. 1 if the first date (parameter) is after the second d) Write a program asking user to for two date and print out the results of the comparison.	10%
5			Solution for exercise 14.1 and explanation	12%
	Exercises 14.2		Write a program that uses a structure to store the following weather data for a particular month: - Total Rainfall - High Temperature - Low Temperature - Average Temperature	10%

			When the program run, it ask the user to enter data for each month and then calculate and display the average rain fall, the total rainfall of the year, the highest and lowest temperatures for the year.	
			Solution for exercise 14.2 and explanation	13%

	Exercises 14.3		Write a student management program using this structure:
			typedef struct
			{
			 char id[6];
			 char name[31];
			 float grade;
			 char classement
			} student;
			The program should read from keyboard data for n students, then print the list of student with the classment in the order descending of grade
				10%

			Solution for exercise 14.3 and explanation	12%
	Exercises 14.4		Define a new type for representing the fractions. Then use it to write a fraction manipulation program. This program has the following functionalities. • Input for an array of fraction • Print the content of the fraction array • Inverse all the fraction in the array • Compare two fraction	10%
			Solution for exercise 14.4 and explanation	13%

Semester: 2 Course Title: C Programming Introduction Week: 15th

Theme: Final Exam

Japanese Expert			HUT Staff	
No.	Keywords	Advice, textbook, etc.	Lecture Scenario	Time req.
1			Final Exam will be taken in the computer room. Students will have about 1 or 2 programming exercises to do.	From 60 to 120 minutes
2				
3				
4				
5				