

KISII UNIVERSITY- KERICHO CAMPUS
FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
COURSE CODE: BINT 0121 UNIT NAME: INTROUCTION TO PROGRAMMING II
Question one [25 marks]

- a. Define the following terminologies
 - i. Pointers (2 marks)
 - ii. Data type (2 marks)
 - iii. List (2 marks)
 - iv. Array (2 marks)
 - v. Stack (2 marks)
- b. Outline the similarities of Structured Programming and Object Oriented Programming (5 marks)
- c. Using an appropriate example, discuss three-dimensional array (5 marks)
- d. List five disadvantages of structured programming (5 marks)

Question two [15 marks]

- a. Declare variables to store the following entities (initialize each variable)
 - i. Student Mark (2 marks)
 - ii. Cost of an Item (2 marks)
 - iii. Alphabetical letter (2 marks)
 - iv. Employee name (2 marks)
 - v. Decimal number (2 marks)
- b. Write a function that returns the area of a triangle (5 marks)

Question three [15 marks]

- a. Discuss three differences between High level programming language and low level programming language (6 marks)
- b. Define datatypes of the following variables
 - a. 34 (1 marks)
 - b. 23.099 (1 mark)
 - c. a (1 mark)
 - d. \$12.99 (1 mark)
 - e. 11/12/2012 (1 mark)

- c. Write a c program to display the message “Welcome to C programming” 20 times. Use either for loop or while loop (4 marks)

Question four [15 marks]

- a. What is the difference between the following:
- i. Break statement and continue statement (4 marks)
 - ii. Local variables and global variables (4 marks)
- b. Briefly discuss any four types of in-built operators that are found in C programming language (7 marks)

Question five [15 marks]

- a. Write a program to prompt a user to enter two number and compares the number and display which of the number is greater than the other. i.e a is greater b (use comments where necessary) (10 marks)
- b. The program below contains syntax errors, rewrite the program and correct the error (Comment where necessary) (5 marks)

```
#include <stdio.h>
int main ()
{
int x, y,z;
z = x+y;
printf(“the sum is:”)
return 1;
}
```

Question six [15 marks]

- a. Using an example, discuss the three types of loops that are used in C (6 marks)
- b. Write a program to calculate the area of a circle, it should prompt the user to inputs radius of the circle in centimeters. Use the formula $A = \pi r^2$ (9 marks)

MARKING SCHEME

Question one

a.

- i. Pointers -in C are easy and fun to learn. Some C programming tasks are performed more easily with pointers, and other tasks, such as dynamic memory allocation, cannot be performed without using pointers.
- ii. Data type - in C refer to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted.
- iii. List - a sequence of links which contains items. Each link contains a connection to another link.
- iv. Array is a kind of data structure that can store a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.
 - i. Stack - is an Abstract Data Type (ADT), commonly used in most programming languages. It is named stack as it behaves like a real-world stack, for example – a deck of cards or a pile of plates, etc.

b.

Structured Programming

- Structured programming follows **top-down approach**.
- Structured Programming is also known as **Modular Programming** and a subset of **procedural programming language**.
- In Structured Programming, Programs are divided into small self contained **functions**.
- Structured Programming is **less** secure as there is no way of **data hiding**.
- Structured Programming can solve **moderately** complex programs.

Object Oriented Programming

- Object oriented programming follows **bottom-up approach**.
- Object Oriented Programming supports **inheritance, encapsulation, abstraction, polymorphism**, etc.
- In Object Oriented Programming, Programs are divided into small entities called **objects**.
- Object Oriented Programming is more secure as having data hiding feature.
- Object Oriented Programming can solve any **complex** programs.

- Structured Programming provides **less reusability**, more function dependency.
- Object Oriented Programming provides more reusability, less function **dependency**.
- Less abstraction and less flexibility.
- More abstraction and more **flexibility**.

c.

- Application programs are easier to read and understand.
- Application programs are less likely to contain logic errors.
- Errors are more easily found.
- Higher productivity during application program development.
- Improved application program design.
- Application programs are more easily maintained.

d.

```
/* function returning the area of a rectangle */
double find_area(double length, double width)
{
/*declaration of area variable*/
Double area;
area = length * width;
return area;
}
```

Question two

a.

- Easy to learn
- Structured language
- It produces efficient programs
- It can handle low-level activities
- It can be compiled on a variety of computer platforms

a.

- Preprocessor Commands
- Functions

- Variables
 - Statements & Expressions
 - Comments
- b.
- i. Using #define preprocessor
#define LENGTH 10
 - ii. Using const keyword
constint LENGTH = 10;
- c.
- i. + addition
 - ii. – subtraction
 - iii. * multiplication
 - iv. % modulus

Question three [15 marks]

- a. A two-dimensional array is, in essence, a list of one-dimensional arrays. To declare a two-dimensional integer array of size [x][y], you would write something as follows:

```
type arrayName [ x ][ y ];

int a[3][4] = {
{0, 1, 2, 3} , /* initializers for row indexed by 0 */
{4, 5, 6, 7} , /* initializers for row indexed by 1 */
{8, 9, 10, 11} /* initializers for row indexed by 2 */

};
```

- b.
- i. Integer – int num1;
 - ii. Double – double num2;
 - iii. Character – char name;
 - iv. Float – float num3;
 - v. Date date today

c.

```
#include<stdio.h>
int main(){
for(inti=0;i<20;i++){
printf(“Welcome to C programming \n”);
}
return 0;
}
```

Question four [

a.

- i. Break statement terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch and continue statement terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch.
- ii. Local variables are variables that are declared inside a function or block are called local variables. They can be used only by statements that are inside that function or block of code and global variables are defined outside a function, usually on top of the program. Global variables hold their values throughout the lifetime of your program and they can be accessed inside any of the functions defined for the program.

b.

- Arithmetic Operators
- Relational Operators
- Logical Operators
- Bitwise Operators
- Assignment Operators
- Misc Operators

Question five

a.

```
#include<stdio.h>
Int main(){
Int a, b, c;
Printf("enter a\n");
Scanf("%d", &a);
Printf("enter b\n");
Scanf("%d", &b);
C= a%b;
Printf("the modulus is: %d", c);
Return 0;
}
```

b.

```
#include <stdio.h>
int main ()
{
int a, b, sum;
printf("enter a and b");
scanf(%d %d, &a, &b);
sum = a + b;
```

```
printf("the sum is: %d, &sum);  
return 0;  
}
```