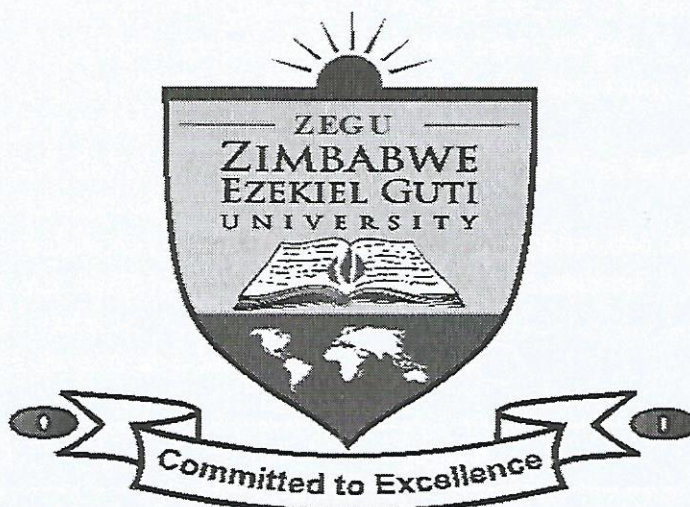


ZIMBABWE EZEKIEL GUTI UNIVERSITY



FACULTY OF HEALTH, SCIENCE AND TECHNOLOGY

Department of Digital Technology

MODULE: INTRODUCTION TO COMPUTER PROGRAMMING
(MAIN)
CODE: CDT 101
DATE: 31 May 2019
DURATION: 4 HOURS

INSTRUCTIONS AND INFORMATION TO CANDIDATES

1. CREATE A FOLDER ON THE DESKTOP AND NAME IT AFTER YOUR REGISTRATION NUMBER. ALL YOUR ANSWERS MUST BE SAVED IN THAT FOLDER, WHICH YOU SHALL SUBMIT AT THE END OF THE EXAMINATION.
2. ATTEMPT ANY FOUR QUESTIONS IN THIS PAPER.
3. THIS IS A PRACTICAL, OPEN BOOK EXAMINATION.
4. ALL PROGRAMS MUST BE WRITTEN IN C or C++.
5. EACH FULL QUESTION CARRIES 25 MARKS.
6. NUMBER OF MARKS FOR EACH QUESTION IS GIVEN IN SQUARE BRACKETS []

Question 1

Nyabiko Private College wants a computer software system to manage their student information and examinations. According to the rules of the college:

- Students are assessed on two components (coursework and final examination) for each course.
- Coursework comprises two (2) assignments and an in-class test, all of which are marked out of a hundred (100). The two assignments and in-class test have equal weight.
- Coursework is worth 30 percent of the final mark, while the final exam, though marked out a hundred, is worth 70 percent of the final mark.
- The final mark for each student for each course is obtained by adding weighted coursework (out of 30) to weighted final examination mark (out of 70).
- For a student to pass a course, they should pass **both** coursework and final examination. If a student passes only one of the two and yet their final mark (coursework + exam mark) is above 50, they get a technical failure.
- The grading scale for the college is as shown in Table 1:

Final Mark	Grade
75-100	Distinction
60-74	Merit
50-59	Pass
Below 50	Fail

Table 1: Grading Scale

Required:

Write a program that meets the requirements of Nyabiko College as specified in the scenario above. Your program should:

- (a) Allow a user to enter three coursework marks (two assignments and in-class test) and final exam mark. All input must be validated such that no mark below zero (0) or above one hundred (100) must be accepted. If the user enters a negative mark or a mark above 100, display an appropriate error message and allow the user to re-enter a valid mark. [10]
- (b) Calculate the weighted coursework mark (out of 30) and weighted exam mark (out of 70) and display them to 1 decimal place. [3]
- (c) Determine the grade for the student and displays it. If the grade is a technical failure, your program should tell the user which component the student failed (coursework or final exam). [10]
- (d) Correct Syntax [2]

Question 2

The equation of a straight line is $y = mx + c$ where m is the gradient. The gradient of a straight line is obtained using the formula $m = (y_2 - y_1)/(x_2 - x_1)$ where x_i and y_i are points on the line.

Required:

Write a program that:

- (a) Uses a function called `accept_values()` to allow a user to enter two coordinates of a straight line [8]
- (b) Uses a function called `calc_gradient()` to calculate the gradient of the line. The function `calc_gradient()` must return a float [6]
- (c) Uses a function called `display()` to display the gradient of the line in a 20 character space, right justified, correct to 3 decimal places. [6]
- (d) Calls the functions in appropriate function(s) to make the program run and display intended output. [3]
- (e) Correct Syntax [2]

Question 3

- (a) Suppose the exchange rate between the USD and RTGS \$ is 1:2.5. To convert USD to RTGS, one would use the formula $rtgs = 2.5 * usd$ where $rtgs$ is the Zimbabwe RTGS dollar and USD is the United States Dollar.

Write a program that converts USD to RTGS using the formula specified above, to produce the output in Table 3. Your program must use a loop, and 2.5 must be defined as a constant. [Hint: No user input is required.] [15]

USD	RTGS Dollar
0	0.00
5	12.50
10	25.00
15	37.50
20	50.00

Table 3 USD to RTGS conversion

- (b) Write a program that uses nested loops to display output similar to the following:

*

**

The number of rows to be displayed should be entered by a user. NB: Your program should only allow a user to enter a positive integer for the number of rows. If the user enters an invalid input, display an appropriate error message and allow the user to re-enter a new value until they enter a valid input. [10]

Question 4

- (a) Write a program that accepts an integers from a user and determines whether it is an even number or not. If the user enters wrong input type, your program should display an appropriate error message and allow the user to re-enter a new value until the input is valid. [6]
- (b) Write a program that calculates an employee's wage based on the formula:
 $wage = hours\ worked * hourly\ rate$ where hourly rate is \$40.00 (forty dollars). The program should accept number of hours worked from a user. The hourly rate must be defined as a constant.
[10]
- (c) Write a program that uses an array of size 5, and loops to accept five (5) integers from a user and display the average, minimum and maximum value. [9]

Question 5

- (a) Write a program that accepts a number or word from a user and determines whether it is a palindrome or not. Hint: A palindrome is a word or number that reads the same forwards and backwards, for example, mum, madam and 110011. [10]
- (b) Write a program that calculates and displays the factorial of a number entered by a user. If the user enters an invalid input, display an appropriate message and allow the user to re-enter a value. [15]

Question 6

- (a) Write a program that allows a user to enter a PIN (Personal Identification Number) up to a maximum of three (3) attempts. If the user enters a wrong PIN three (3) times, display the message "Sorry, your account has been locked because you entered a wrong PIN three times. Good bye!" If the user enters the correct PIN in three attempts, display the message, "Welcome. You entered the correct PIN". Assume the correct PIN is 1234. [10]
- (b) Write a program that does the following:
 - (i) Creates a structure called **Employee**, with the following members - first name, surname, date of birth, position, salary. [6]
 - (ii) Creates an array of type **struct Employee** of size 3 to store details of three employees. [1]
 - (iii) Creates a file called **employees.txt** which must be saved in the folder you created on the desktop, named after your registration number [3]
 - (iv) Allows a user to enter details of three (3) employees and stores them in the file called **employees.txt** that you created above. [5]

***** End of Paper *****