



**NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF APPLIED SCIENCES
DEPARTMENT OF INFORMATICS AND ANALYTICS
PROGRAMMING CONCEPTS AND DEVELOPMENT USING
PYTHON**

SIA 1204

**MAIN EXAMINATION PAPER
SECOND SEMESTER 2024**

This examination paper consists of 7 pages.

Time Allowed: 3 hours
Total Marks: 100
Examiner's Name: Mrs C. Chivasa

INSTRUCTIONS

1. Answer all questions in Section A and any 3 in Section B
2. Section A carries 40 marks.
3. Section B carries 60 marks.
4. Use of calculators is permissible.

MARK ALLOCATION

QUESTION	MARKS
Section A	
1.	40
Section B	
2.	20
3.	20
4.	20
5.	20

SECTION A

QUESTION ONE

a) **Pizza Party Panic!**

It's Friday night, and you're having a huge pizza party with your friends. You've ordered a variety of pizzas with different toppings to satisfy everyone's taste. But there's a problem you can't remember who ordered what!

This is where your python programming skills come in handy to ensure everyone gets the perfect slice.

Challenge 1: The Order Master

To avoid future confusion, you decide to create a python program to manage pizza orders.

- Write a python program that allows users to enter their name and desired pizza toppings. The program should store this information in a list or dictionary for easy reference. **[10]**

Challenge 2: The Slice Savior

With the order list in hand, you need to figure out how to distribute the pizza slices fairly. Some pizzas might have more slices than others, and some friends might prefer specific toppings.

- Write a python program that takes the order list and the number of slices per pizza as input? The program should then create a recommendation for who gets which slice based on their preferences and ensuring everyone gets a (somewhat) equal share. (Hint: List manipulation and conditional statements) **[10]**

Challenge 3: The Leftover Lament

After the party, there are leftover slices! You don't want them to go to waste, so you decide to create a "Leftover Lover" program.

- Write a python program that tracks leftover slices and their toppings? It should allow users to search for specific leftover combinations or suggest recipes that could utilize the remaining ingredients. (Hint: Dictionaries with nested structures and string manipulation) [12]

b) Write a program that does the following in order:

i. Asks the user to enter a number “x”

ii. Asks the user to enter a number “y”

iii. Prints out number “x”, raised to the power “y”.

iv. Prints out the log (base 2) of “x”.

[8]

SECTION B

QUESTION TWO

You have graduated from NUST and now have a great job! You move to the Harare and decide that you want to start saving to buy a house. As housing prices are very high, you realize you are going to have to save for several years before you can afford to make the down payment on a house. We are going to determine how long it will take you to save enough money to make the down payment given the following assumptions:

1. Call the cost of your dream home `total_cost`.
2. Call the portion of the cost needed for a down payment `portion_down_payment`. For simplicity, assume that `portion_down_payment = 0.25` (25%).
3. Call the amount that you have saved thus far `current_savings`. You start with a current savings of \$0.

4. Assume that you invest your current savings wisely, with an annual return of r (in other words, at the end of each month, you receive an additional $\text{current_savings} * r / 12$ funds to put into your savings – the 12 is because r is an annual rate). Assume that your investments earn a return of $r = 0.04$ (4%).

5. Assume your annual salary is annual_salary .

6. Assume you are going to dedicate a certain amount of your salary each month to saving for the down payment. Call that portion_saved . This variable should be in decimal form (i.e. 0.1 for 10%).

7. At the end of each month, your savings will be increased by the return on your investment, plus a percentage of your monthly salary ($\text{annual_salary} / 12$).

Write a program to calculate how many months it will take you to save up enough money for a down payment. You will want your main variables to be floats, so you should cast user inputs to floats.

Your program should ask the user to enter the following variables:

1. The starting annual salary (annual_salary)

2. The portion of salary to be saved (portion_saved)

3. The cost of your dream home (total_cost) [20]

QUESTION THREE

a) Identify any 5 keywords that are used in python. [5]

b) Write a python program to check whether the string is a symmetrical or palindrome. [10]

c) $L = 3$

```
for i in range(len(L)):
    print(i)
```

ERROR MESSAGE:

```
File "C:/Users/Ana/.spyder2-py3/temp.py", line 2, in
for i in range(len(L)):
```

TypeError: object of type 'int' has no len()

Above is a piece of code and an error shown when running it. What is the problem? [5]

QUESTION FOUR

a) Write a program that will prompt the user for an age in dog years and calculate that age in human years. Allow for int or float values, but check the user's input to make sure it's valid. It should be numeric and positive. Otherwise, let the user know their input is not valid.

You can use the following rules to approximately convert a medium-sized dog's age to human years:

For the first year, one dog year is equal to 15 human years

For the first 2 years, each dog year is equal to 12 human years

For the first 3 years, each dog year is equal to 9.3 human years

For the first 4 years, each dog year is equal to 8 human years

For the first 5 years, each dog year is equal to 7.2 human years

After that, each dog year is equal to 7 human years. (Note: This means the first 5 dog years are equal to 36 human years ($5 * 7.2$) and the remaining dog years are equal to 7 human years each.)

Print the result in the following format, substituting for `<dog_age>` and `<human_age>`:

"The given dog age `<dog_age>` is `<human_age>` in human years." Round the result to 2 decimal places. Note: If there is a 0 in the hundredths place, you can drop it, e.g. 24.00 can be displayed as 24.0.

For example:

If the user enters 2, the program will print: "The given dog age 2 is 24.0 in human years."

If the user enters 3, the program will print: "The given dog age 3 is 27.9 in human years."

If the user enters 4.5, the program will print: "The given dog age 4.5 is 32.4 in human years."

If the user enters 12.1, the program will print: "The given dog age 12.1 is 85.7 in human years."

Considering invalid inputs:

Your program must ask the user for an age in dog years - hint: use the `input()` function

We are going to test invalid inputs - make sure that your code can handle negative value inputs and non-numerical inputs!

For invalid inputs, make sure that your printed response adheres to the following:

If a text-based input is provided, make sure your response contains the word '**invalid**'.

For example, if the user doesn't input a number, print "<age> is invalid.", substituting for <age>.

If a negative input is provided, make sure your response contains the word '**negative**'.

For example, if the user inputs a negative number, print "Age cannot be a negative number."

[20]

QUESTION FIVE

a) Explain the following OOP terms

i Polymorphism

[3]

ii Inheritance

[4]

iii Encapsulation

[3]

b) Explain any three programming constructs using code snippets.

[10]

END OF QUESTION PAPER