**Assignment Content**

1. **Experiments**
2. Evaluate following and express the results:

5 > 2, 5 == 2, 5=2.

1. Determine whether the given number is even or odd.
2. Marks of a student in five subjective are acquired. Determine average marks and grade. O - 91-100 A+ 81 -90 A 71 - 80 B+ 61-70 B 51 – 60 C+ 41 - 50 C 35 - 40 Fails otherwise.
3. Evaluate 15 & 27 and verify your result.
4. Evaluate 15 | 27 and verify your result.
5. Evaluate 15 ^ 27 and verify your result.
6. Multiply 3 by 4 using shift bitwise operator.
7. Divide 64 by 8 using shift bitwise operator.
8. **Experiments**
9. WAP add to numbers.
10. WAP maximum of 2 numbers.
11. WAP find factorial of a number.
12. WAP find simple interest.
13. WAP find prime number set.
14. WAP to check if a number is prime or not.
15. **Experiments**
16. Generate squares of all the integers from 1 to 50.
17. Count the number of characters in a string using a loop.
18. Print a string in reverse.
19. Find all the prime numbers below 50.
20. Sum all the multiple integers of 5 below 50.
21. Generate the patterns given below:

|  |  |  |
| --- | --- | --- |
| \*  \* \*  \* \* \* \*  \* \* \* \* \* | 1  2 1 2  3 2 1 2 3  4 3 2 1 2 3 4 | 1 2 3 4 5  2 3 4 5  3 4 5  4 5  5 |
|  |  |  |

1. Print Armstrong numbers in the range 1 to 1000. An Armstrong number is a number whose sum of the cubes of the digits is equal to the number itself. For example, 370 = 33+73+03.
2. Create an employee database having record for 5-10 employees in it. The attributes for an employee record are his/her name, age, salary, address. Take inputs from the user.

* Print each record on a separate line
* Update a record with given name.

1. **Experiments**
2. Use following predefined functions and interpret the results:

Min () and max ()

Bin (), Oct (), Hex ()

Pow ()

Eval () and exec ()

Chr () and Ord ()

Round ()

Random (), rand int (), Import random module Random.rondom () Import math module

Sin (), Cos (), Tan (), Ceil (), Floor (), Degrees ().

1. Write a function to sort the contents of an integer list.
2. Write a function to change the case of a given string.
3. The Fibonacci Sequence is the series of numbers: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Write a function recursive) to print n terms of this series based upon user input.
4. **Experiments**
5. Write a Python program to use try-except-else block.
6. Write a Python program to use try-except-finally block.
7. Write a Python program to write your own exception and throw it.
8. Write a Python program to demonstrate the use of built-in modules random and math.
9. Write a program to create a module and to use its functionality.
10. **Experiments**
11. Write a program to count the number of English alphabets (lowercase and uppercase) in a text file.
12. Write a program to count the number of lines in a text file.
13. Write a program to count the occurrences of a specific word in a text file.
14. Write a Python program to take input from the user and record it in a file. Once the input is over, program also shows the content of a file.
15. Create a small database of employees that contains following attributes: Name, age, salary. Write a menu-driven program to

* List all the employees
* appending a new employee
* Searching for a specific employee

1. **Experiments**
2. Write a class named Rectangle. Take the appropriate attributes and methods. Instantiate this class to use its attributes.
3. Create a class Circle with attributes centre and radius. Add methods for area and circumference calculation. Now create circle objects to use these methods. Give the provision to take circle attributes from the user.
4. Create a class named Account. An Account object can have following attributes:

Account Number

Account Holder

Account Type

Balance

credit\_account()

debit\_account()

get\_interest() (SI)

get\_account()

set\_account()

Add some class attributes to this Account class. Provide the appropriate constructor.

Create some sample bank account, use the listed methods and finally delete them.

Dynamically create account objects by navigating through a menu, such as:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create account

Withdraw amount

Deposit amount

Show Interest

Delete account

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter your choice?

1. **Experiments**
2. Write a class named Rectangle. Take length and breadth as attributes and area as methods. Create a subclass Square with the only attribute as side and override the area method. Create Rectangle and Square object statically and dynamically and make use of area method.
3. Overload the + (addition) operator to restrict the addition of two integers to modulo 8.
4. Implement a Stack class using a list type. Provide push() and pop() operations. Demonstrate the usage.
5. **Experiments**
6. Write your own iterator class and iterate through it (without generator function).
7. Write your iterator by using generator function.
8. Write a function that calculates simple interest. Now decorate the output of this function by returning the modified amount.