20ES2103A

OBJECT ORIENTED PROGRAMMING USING PYTHON

INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

1. Differentiate between a procedural language and an object-oriented language.
2. Explain the main features of an object-oriented programming language.
3. Differentiate between a class and an object.
4. Inheritance helps to make reusable code. Justify.
5. What do you understand by the term 'polymorphism?
6. Explain the three levels of data protection in OOP.
7. What are the merits and demerits of OOPs?
8. List out merits, demerits and applications of object-oriented programming language.
9. Elaborate Data abstraction and encapsulation with a real-world example.
10. How containership is different from delegation?

BASICS OF PYTHON PROGRAMMING

1. List out the applications of Python scripting language.
2. Summarize the features and limitations of Python.
3. Differentiate arithmetic overflow, arithmetic underflow and loss of Precision Problems.
4. Define Python identifiers, variables and data types with suitable examples.
5. Explain different types of operators supported by Python language.
6. Classify the Python operators based on their precedence and associativity.
7. What is an expression, and recall different types of expressions based on the position of operators in an expression?
8. How type casting is different from type coercion? Justify your answer with proper examples.
9. How + and * operators work differently on string data than integer and float data?
10. What is slicing operator? How can you extract a substring from a given string?
11. Explain the significance of escape sequences with the help of relevant examples.
12. Outline implicit and explicit type conversions? Explain the need for type conversion with the help of relevant examples.
13. Write a Python program to enter two integers and then perform all arithmetic operations on them. Repeat the program using floating point numbers.
14. Write Python programs to print ASCII value of a character and character from an ASCII value.
15. Write a Python program to swap two numbers using a temporary variable.
16. Write a Python program to calculate simple interest and compound interest.
17. Write a program that prompts user to enter his first name and last name and then displays a message "Greetings!!!First name Last name".

18. Write a program to calculate gross salary of an employee given his basic pay (to be entered by the user). HRA = 10% of basic pay, DA = 80% of basic pay. Gross salary = basic pay + basic pay *(HRA+DA).
19. Momentum is calculated as, $e = mc^2$, where m is the mass of the item and c is its velocity. Write a program that accepts an item's mass (in Kgs) and velocity (in m/s) and displays its momentum.
20. Develop a Python program to take three float numbers as sides of a triangle and print the area of a triangle.
21. Write a Python program to calculate area and perimeter of a circle.

DECISION CONTROL STATEMENTS

1. Define decision control statements. Explain different types of selection control statements with proper syntaxes.
2. Compare nested if-else statement with if-elif-else statement. Develop separate Python programs using the nested if-else and if-elif-else statements to check a year is a leap year or not.
3. Write Python programs to find smallest integer number between three integer numbers using nested if else and if-elif-else statement.
4. Write a Python program to take three integer numbers (perpendicular, base, and hypotenuse) and print whether these numbers form a right-angle triangle or not.
5. Develop a Python program to take three integer numbers as sides of a triangle and verify whether these numbers will form a triangle or not. If so, then print the area of a triangle.
6. Write a program to enter a character to check whether it is a vowel or a consonant. If the character is in lowercase, convert it into uppercase, and if it is an uppercase character, convert it into lowercase.
7. A company decides to give bonus to all its employees on Diwali. A 5% bonus on salary is given to the male workers and 10% bonus on salary to the female workers. Write a program to enter the salary of the employee and sex of the employee. If the salary of the employee is less than 10000/- then the employee gets an 2% bonus on salary. Calculate the bonus that has to be given to the employee and display the salary that the employee will get.
8. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, and display the grade obtained by the student. (aggregate > 75% : Distinction, 60% <= aggregate < 75% : First Division, 50% <= aggregate < 60% : Second Division, 40% <= aggregate < 50% : Third Division, Else Fail).
9. Write a Python program to calculate real roots of a quadratic equation.
10. Write a Python program to calculate parking charges of a vehicle. Enter the type of vehicle as a character (like c for car, b for bus, etc.) and number of hours, then calculate charges as given below:
    Truck/bus - 20/- per hour, Car – 10/- per hour, Scooter/Cycle/Motor cycle – 5/- per hour

11. Develop a Python program that determines whether a digit, uppercase or a lowercase character was entered.
12. For loop is usually known as a determinate or definite loop. Justify the statement with the help of an example.
13. With the help of an example, explain the utility of range().
14. Differentiate between counter-controlled loops and sentinel-controlled loops.
15. Explain the utility of break, continue and pass statements with the help of an example.
16. Do a comparison between the pre-test and post-test loops. Briefly explain different types of pre-test loops with suitable examples in Python.
17. Write a Python program to sum the series $1^2/1 + 2^2/2 + 3^2/3 + \ldots + n^3/n$ using for loop.
18. Write a Python program that prints all the prime numbers in the range (1, n).
19. Write a program that prompts users to enter numbers. Once the user enters –1, it displays the count, sum, and average of even numbers and that of odd numbers.
20. Write a Python program to print prime factors of a number.
21. Write a program to display the cos(x) and sin(x) value where x ranges from 0 to 360 in steps of 15.
22. Write an interactive program to read an integer. If it is positive then display the corresponding binary representation of that number. The user must enter 999 to stop. In case the user enters a negative number, then ignore that input and ask the user to re-enter any different number.
23. Write a program that accepts any number and prints the number of digits in that number. Expand the program to print the reverse of that number.
24. Write a simple Python program that displays the following powers of 2. one per line: 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8.
25. Write a Python program using while loop to calculate factorial of a number.
26. Write a Python program to print Fibonacci series up to step n.
27. Make use of while loop to develop a Python program to check a number is a strong number or not.
28. Develop a Python program to check a number is a perfect number or not using while loop.
29. Develop a Python program to check a number is an Armstrong number or not using for loop.
30. Write a Python program to calculate GCD of two integer numbers.
31. Write python programs to print the following patterns.

i.

```
1

2 2

3 3 3

4 4 4 4

5 5 5 5 5
```

ii.
```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

32. Write Python programs to print the following patterns.

i.
```
*
* *
* * *
* * * *
* * * * *
```

ii.
```
    *
   * *
  * * *
 * * * *
* * * * *
```

33. Write Python programs to print the following patterns.

i.
```
        1
      1 2
    1 2 3
  1 2 3 4
1 2 3 4 5
```

ii.
```
1 2 3 4 5
  1 2 3 4
    1 2 3
      1 2
        1
```