1.SymPy is a collection of mathematical algorithms and convenience functions built on the -----------extension of Python

|  |  |
| --- | --- |
| **a)numpy** | b)scikit |
| c)sys | d) functools |

2. Exponential function computes the ----------------

|  |  |
| --- | --- |
| **a) 10\*\*x element-wise** | b) 10\*\*x row-wise |
| c) 10\*\*x column-wise | d) 10\*x element-wise |

3.\_\_\_\_\_\_\_\_\_\_\_ evaluates the expression to a floating-point number.

|  |  |
| --- | --- |
| **a)evalf** | b)fval |
| c) float | d) valf |

4. what is the output for the following expression ((x+y)\*\*2).expand()

|  |  |
| --- | --- |
| a) x,2 + 2,x,y + y,2 | b) x\*2 + 2\*x\*y + y\*2 |
| c) x^2 + 2\*x\*y + y^2 | **d) x\*\*2 + 2\*x\*y + y\*\*2** |

5. limit((5\*\*x + 3\*\*x)\*\*(1/x), x, oo) ,what is the output

|  |  |
| --- | --- |
| **a)5** | b) oo |
| c) 1 | d) 0 |

6) Higher derivatives can be calculated using the which method

|  |  |
| --- | --- |
| a) highder(func,var,n) | **b) diff(func, var, n)** |
| c) diff(n,var,func) | d) diff(func, var) |

7) what is the output

>>> x = Symbol('x')

>>> y = Symbol('y')

>>> A = Matrix([[1,x], [y,1]])

>>> A\*\*2

|  |  |
| --- | --- |
| a) [1, x]  [y, 1] | b) [xy + 1, 2x]  [2y, xy + 1] |
| c) [x\*y + 1, 2\*y]  [2\*x, x\*y + 1] | **d) [x\*y + 1, 2\*x]**  **[ 2\*y, x\*y + 1]** |

8) .match() method, along with the ------- class, to perform pattern matching on expressions.

|  |  |
| --- | --- |
| a) pattern | b) func |
| **c) wild** | d) dictionary |

9) which among the following function Return or print, respectively, a pretty representation of expr

|  |  |
| --- | --- |
| a) pretty(expr) | b) pretty\_print(expr) |
| c) pprint(expr) | **d) all of the above** |

10) What is the output of

from sympy.abc import a, b

expr = b\*a + -4\*a + b + a\*b + 4\*a + (a + b)\*3

|  |  |
| --- | --- |
| a) ba-4a+b+ab+4a+3(a+b) | **b) 2\*a\*b + 3\*a + 4\*b** |
| c) 2ab+3a+4b | d) all of the above |

11) print(pi.evalf(30))

|  |  |
| --- | --- |
| a) 3.14/30 | b)30/3.14 |
| **c)3.14159265358979323846264338328** | d) 3.14 |

12) which is the correct way to write equation for x^2=x in sympy

|  |  |
| --- | --- |
| **a) x\*\*2 =x** | b) x\*x = x |
| c) x%2 = x | d) X^2 =x |
|  |  |

13) how to find a solution for a equation in a given interval

|  |  |
| --- | --- |
| a) solve(equation,range) | b)equation(solve,range) |
| **c) solveset()** | d) both a and b |
|  |  |

14) Allows , the same elements can appear multiple times at different positions

|  |  |
| --- | --- |
| a) set | **b)sequence** |
| c) dictionary | d) none |

15) which is the snippet to find the eigenvalues of [1 2

2 2]

|  |  |
| --- | --- |
| **a) Matrix([[1, 2], [2, 2]]).eigenvals()** | b) Matrix([[1, 2], [2, 2]]).eigen |
| c) both a and b | d) eigen(([[1, 2], [2, 2]]) |

16 What is the purpose of sympify() method?

1. **Convert expression of string type to mathematical expression**
2. Convert mathematical expression to String
3. Convert mathematical expression to character
4. Convert tuple to mathematical expression

17 Find the output of the following program

**from sympy import solve** **x = Symbol('x')** **expr = x\*\*2 + 5\*x + 4  
solve(expr, dict=True)**

1. **[{x: -4}, {x: -1}]**
2. [{x: -6}, {x: -1}]
3. [{x: -1}, {x: -4}]
4. [{x: 4}, {x: -1}]

18. A *rational expression* is an algebraic expression in which the numerator and denominator are both --------------

1. Equal
2. **Polynomials**
3. Unequal
4. Symmetric

19. Find the output of the following program

# import sympy

from sympy import \*

x = symbols('x')

expr = sin(x)/x;

print("Expression : {}".format(expr))

# Use sympy.limit() method

limit\_expr = limit(expr, x, 0)

print("Limit of the expression tends to 0 : {}".format(limit\_expr))

1. Expression : cos(x)/x

Limit of the expression tends to 0 : 2

1. Expression : tan(x)/x

Limit of the expression tends to 0 : 3

1. Expression : sin(x)/x

Limit of the expression tends to 1 : 0

1. **Expression : sin(x)/x**

**Limit of the expression tends to 0 : 1**

20. -------------- method will simplify mathematical expression using trigonometric identities.

1. **sympy.trigsimp()**
2. sympy.series()
3. sympy.lambda()
4. sympy.sim()