Q.1. Python files are saved with the extension as …?

a. .python

b. .pe

c. .py

d. .pi

Answer:- .py

Q.2. What is the name of the GUI that comes in-built as an interactive shell with Python?

a. PGUI

b. Pyshell

c. IDLE

d. PythonSh

Answer:- PGUI

Q.3. IDLE stands for … ?

a. Indigenous Development Lab

b. Integrated Development Environment

c. Integrated Developers Local Environment

d. Indie Developers Environment

Answer:- integrated developer local environment.

Q.4. Which of the following is an assignment operator in Python?

a. ==

b. ===

c. >>>

d. =

Answer:- d.=

Q.5. A user-specified value can be assigned to a variable with this function …

a. user

b. enter

c. input

d. value

Answer: - b.input

Q.6. User input is read as …?

a. Floating Decimal

b. Text String

c. Boolean Value

d. Integer

Answer: -b.text string

Q.7. What will be the output after the following statements?

x = 6

y = 3

print(x / y)

a. 2.0

b. 2

c. 18

d. 18.0

Answer:- a. 2.0

Q.8. What will be the data type of x after the following statement if input entered is 18 ? x = input(‘Enter a number: ‘)

a. Float

b. String

c. List

d. Integer

Answer:- b.string

Q.9. What will be the data type of y after the following statements?

```
x = 71

y = float(x)
```

a. Float

b. String

c. List

d. Integer

Answer:- a. Float

Q.10. What is the data type of x after the following statement?

x = [7, 8, 9, 10]

a. List

b. Dictionary

c. Tuple

d. String

Answer:- a. List

Q.11. Which of the following does not correctly create an object instance?

A. puppy = Dog("Jamie")

B. dog = Dog("Jamie")

C. jamie = Dog()

D. pupper = new Dog("Jamie")

Answer:- d. new dog(“jamie”)

Class name doesn't support blank spaces

Q12. Which of the following is required to create a new instance of the class?

A.constructor

B. class

C. value-returning method

D. None method

Answer:- b. class

Q13. Which of the following statements is most accurate for the declaration x = Circle()?

A.x contains an int value.

B.x contains an object of the Circle type.

C.x contains a reference to a Circle object.

D.You can assign an int value to x.

Answer:- c. X contains a refrence to a circle object.

Q14. Create a Cricle class and intialize it with radius. Make two methods getArea and getCircumference inside this class.

Answer:-

```
class circle:
    def getarea(self,area,result):
        self.area=area
        self.result=result
        result=2(3.14*self.area)
    def getcircumference(self,circumfrence):
        self.circumfrence=circumfrence
    def display(self):
        print(self.result)
obj1=circle()
obj1.getarea(5)
obj1.getcircumfrence()
obj1.display()
```

Q15.Create a Temprature class. Make two methods :

1. convertFahrenheit - It will take celsius and will print it into Fahrenheit.
2. convertCelsius - It will take Fahrenheit and will convert it into Celsius

Answer:-

```
class temprature:

    def convertfahrenheit(self):
       self.cel=int(input('enter temprature in celsius'))
       self.formula= (self.cel * 1.8) + 32
       print("your answer =",self.formula)



    def convertcelsius(self):
        self.fah=int(input("enter temprature in fahrenheit"))
        self.formula= (self.fah -32) * 0.555
        print(" your answer =",self.formula)



obj1=temprature()
obj1.convertfahrenheit()
obj1.convertcelsius()
```

Q16.Create a Student class and initialize it with name and roll number. Make methods to :

1. Display - It should display all informations of the student.
2. setAge - It should assign age to student
3. setMarks - It should assign marks to the student.

Answer:-

```
class student:

    def __init__(self):
        self.name=input("enter name :")
        self.roll_no=input("enter roll no")
        self.dic1={'name':self.name,'roll_no' :self.roll_no}

    def display(self):
       print(self.dic1)
    def setage(self):
        self.age=int(input("enter age :"))
        self.dic1['age']=self.age
    def setmarks(self):
        self.marks=int(input("enter marks :"))
        self.dic1['marks']=self.marks
obj1=student()
obj1.display()
obj1.setage()
obj1.setmarks()
obj1.display()
```

Q17.Create a Time class and initialize it with hours and minutes.

1. Make a method addTime which should take two time object and add them. E.g.- (2 hour and 50 min)+(1 hr and 20 min) is (4 hr and 10 min)

2. Make a method displayTime which should print the time.

3. Make a method DisplayMinute which should display the total minutes in the Time. E.g.- (1 hr 2 min) should display 62 minute.

Answer:-

```
class Time:

    def addtime(self):
        self.hr=int(input("enter hours :"))
        self.min=int(input("enter minute :"))

    def displaytime(self):
        print("{}:{}".format(self.hr,self.min))

    def displayminute(self):

        self.minute=(self.hr*60) +self.min
        print(" Totle minute in this time = ",self.minute)

obj1=Time()
obj1.addtime()
obj1.displaytime()
obj1.displayminute()
```

Q18. Implement Stack using Switch Case

Answer:-

```
class user_difine_stack_oprations:

    def __init__(self,initial_stack):
        self.initial_stack=initial_stack

    def stack_isempty(self):
        if len(self.initial_stack)==0:
            print("-----> STACK IS EMPTY \n")
        else:
            print("-----> STACK IS NOT EMPTY\n")

    def push_elements(self,value_for_push):
        self.value_for_push=value_for_push
        self.initial_stack.append(value_for_push)
        print("-----> SUCCESSFULLY OPRATION PUSH\n")

    def display_stack(self):
        print("\n-----> UPDATED STACK : \n")
        for x in reversed(self.initial_stack):
            print(" ",x)

    def pop_element(self,enter):
        self.enter=enter
        if enter=='1':
            self.initial_stack.pop()
            print("-----> ELEMENT POP SUCCESSFULLY ")
        else:
            print("-----> YOU ENTERED WRONG NUMBER ")

stack1=['dolar','rajesh','jain','paddhariya']
obj1=user_difine_stack_oprations(stack1)
```

```
while True:
 print("-----> ENTER YOUR CHOICE :")
 choice=input("\n1.DISPLAY STACK\n2.STACK EMPTY?\n3.PUSH\n4.POP\n0.EXIT\n-----> choice :")
 if choice=='1':
   obj1.display_stack()
 elif choice=='2':
   obj1.stack_isempty()
 elif choice=='3':
   obj1.push_elements(value_for_push=input("-----> ENTER value YOU WANT TO PUSH :"),)
 elif choice=='4':
   obj1.pop_element(enter=input("-----> ENTER 1 IF YOU WANT TO POP ELEMENT FROM STACK :"))
 elif choice=='0':
   break
```

Q19. Implement Queue Using Switch Case

Answer:-

```
class Qoperations:

  def __init__(self,initial_queue):
    self.initial_queue=initial_queue

  def addQ(self):
    self.value=value=input("\nENTER VALUE FOR ADD ELEMENTS :")
    return self.initial_queue.append(value)

  def removeQ(self):
    if len(self.initial_queue)!=0:
      return self.initial_queue.pop(0)
    else:
            print("\nQueue Is Empty Already you can't remove element ")


    def displayQ(self):
        for item in self.initial_queue:
            print(item,end=" ")
        print("\n")
    def Qisempty(self):
        if len(self.initial_queue)==0:
            print("\nQueue is Empty")
        else:
            print("\nQueue is Not Empty")



Q1=[10,20,30,40,50,]
Queue1=Qoperations(Q1)
while True:
    print("\n1.Check Queue is Empty ?\n2.Add Elements to Queue\n3.Remove Elements from Queue\n4.Display Queue\n0.EXIT.")
    choice=input("\nEnter your choice :")
    if choice=='1':
        Queue1.Qisempty()
    elif choice=='2':
        Queue1.addQ()
    elif choice=='3':
        Queue1.removeQ()
    elif choice=='4':
        Queue1.displayQ()
    elif choice=='0':
        break
```