Data Structures & Algorithms using Python

Quiz Answers

OOPs-3

Name : **Harsh Siddhapura**

Degree : **Bachelor of Technology**

Dept. : **Information & Communication Technology**

What will be the output of the following code?
from abc import ABC,abstractmethod

```
class A(ABC):

    @abstractmethod
    def fun1(self):
        pass

    @abstractmethod
    def fun2(self):
        pass

o = A()
o.fun1()
```

Nothing will be printed.
Error
None of the above

What will be the output of the following code?
from abc import ABC,abstractmethod

```
class A(ABC):

    @abstractmethod
    def fun1(self):
        pass

    @abstractmethod
    def fun2(self):
        pass

class B(A):

    def fun1(self):
        print("function 1 called")
o = B()
o.fun1()
```

function 1 called
Nothing will be printed.
Error
None of the above

What will be the output of the following code?
from abc import ABC,abstractmethod

```
class A(ABC):

    @abstractmethod
    def fun1(self):
        pass

    @abstractmethod
    def fun2(self):
        pass

class B(A):

    def fun1(self):
        print("function 1 called")

    def fun2(self):
        print("function 2 called")
o = B()
o.fun1()
```

function 1 called
Nothing will be printed.
Error
None of the above

What will be the output of the following code?
from abc import ABC,abstractmethod

```
class A(ABC):

    @abstractmethod
    def fun1(self):
        print("function of class A called")

    @abstractmethod
    def fun2(self):
        pass

class B(A):
    def fun1(self):
        print("function 1 called")
    def fun2(self):
```

Harsh Siddhapura   **Page : 3**

```
    print("function 2 called")
o = B()
o.fun1()
```

function 1 called
function of class A called
Error
None of the above

What will be the output of the following code?
```
from abc import ABC,abstractmethod

class A(ABC):

    @abstractmethod
    def fun1(self):
        print("function of class A called")

    @abstractmethod
    def fun2(self):
        pass

class B(A):
    def fun1(self):
        super().fun1()
    def fun2(self):
        print("function 2 called")
o = B()
o.fun1()
```

function 2 called
function of class A called
Error
None of the above

What will be the output of the following code?
```
try:
    a = 10
    b = 0
    c = a/b
    print(c)
except ValueError:
    print("Exception occured")
```

Exception occured
Value Error
ZeroDivisionError
None of the above

What will be the output of the following code?
```
try:
    a = 10
    b = 0
    c = a/b
    print(c)
except ZeroDivisionError:
    print("Exception occured")
```

Exception occured
ValueError
ZeroDivisionError
None of the above

What will be the output of the following code?
```
try:
    a = 10
    b = 0
    c = a/b
    print(c)
except :
    print("Exception occured")
```

Exception occured
ValueError
ZeroDivisionError
None of the above

What will be the output of the following code?
```
try:
    a = 10
    b = 0
    print(d)
    c = a/b
except NameError:
    print('Name Error occured')
except ZeroDivisionError:
```

Harsh Siddhapura   **Page : 5**

```
    print('Zero Division Error occured')
```

Zero Division Error occured
Name Error occured
Error
None of the above

What will be the output of the following code?
```
class ZeroDenominatorError(Exception):
    pass
try:
    a = 10
    b = 0
    if(b==0):
        raise ZeroDenominatorError()
    c = a/b
except ZeroDivisionError:
    print('Zero Division Error occured')
```

Zero Division Error occured
Error - Zero Division Error
Error - ZeroDenominatorError
None of the above

What will be the output of the following code?
```
class ZeroDenominatorError(ZeroDivisionError):
    pass
try:
    a = 10
    b = 0
    if(b==0):
        raise ZeroDenominatorError()
    c = a/b
except ZeroDivisionError:
    print('Zero Division Error occured')
except ZeroDenominatorError:
    print('Zero Denominator Error occured')
```

Zero Division Error occured
Zero Denominator Error occured
Error - ZeroDenominatorError
None of the above

What will be the output of the following code?

```
class ZeroDenominatorError(ZeroDivisionError):
    pass
try:
    a = 10
    b = 0
    if(b==0):
        raise ZeroDenominatorError()
    c = a/b
except ZeroDivisionError:
    print('Zero Division Error occured',end= ' ')
except ZeroDenominatorError:
    print('Zero Denominator Error occured',end = ' ')
else:
    print('else works')
```

Zero Division Error occured else works
Zero Denominator Error occured else works
Zero Division Error occured
Zero Denominator Error occured

What will be the output of the following code?

```
class ZeroDenominatorError(ZeroDivisionError):
    pass
try:
    a = 10
    b = 5
    if(b==0):
        raise ZeroDenominatorError()
    c = a/b
except ZeroDivisionError:
    print('Zero Division Error occured',end= ' ')
except ZeroDenominatorError:
    print('Zero Denominator Error occured',end = ' ')
else:
    print('else works')
```

Zero Division Error occured else works
Zero Denominator Error occured else works
else works
None of the above

What will be the output of the following code?

```python
class ZeroDenominatorError(ZeroDivisionError):
    pass
try:
    a = 10
    b = 5
    if(b==0):
        raise ZeroDenominatorError()
    c = a/b
except ZeroDivisionError:
    print('Zero Division Error occured',end= ' ')
except ZeroDenominatorError:
    print('Zero Denominator Error occured',end = ' ')
else:
    print('else works',end=' ')
finally:
    print('finally works')
```

Zero Division Error occured else works finally works
Zero Denominator Error occured else works finally works
else works finally works
finally works