Snippet 3

Total points 11/30

✗ What will be the output of the following program? ···/1

```
interface One
{
    String s = "FINAL";

    String methodONE();
}

interface Two
{
    String methodONE();
}

abstract class Three
{
    String s = "NOT FINAL";

    public abstract String methodONE();
}

class Four extends Three implements One, Two
{
    public String methodONE()
    {
        String s = super.s + One.s;

        return s;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        Four four = new Four();

        System.out.println(four.methodONE());

        One one = four;

        System.out.println(one.s);
    }
}
```

NOT FINAL FINAL NOT FINAL FINAL ✗

···/1

Can you find out the errors in the following code?

```
interface A
{
    {
        System.out.println("Interface A");
    }

    static
    {
        System.out.println("Interface A");
    }
}
```

can't given defination

0/1

```
interface ABC
{
    void methodOne();
}

interface PQR extends ABC
{
    void methodTwo();
}

abstract class XYZ implements PQR
{
    public void methodOne()
    {
        methodTwo();
    }
}

class MNO extends XYZ
{
    public void methodTwo()
    {
        methodOne();
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        ABC abc = new MNO();

        abc.methodOne();
    }
}
```

- (●) Compile Time Error
- ( ) Run Time Error

Correct answer

- (●) Run Time Error

✗ ···/1

) How do you access interface field 'i' in the below code?

```
class P
{
    interface Q
    {
        int i = 111;
    }
}
```

✗

Correct answer

P.Q.i

0/1

What will be the output of the following program?

```
interface A
{
    void myMethod();
}

class B
{
    public void myMethod()
    {
        System.out.println("My Method");
    }
}

class C extends B implements A
{

}

class MainClass
{
    public static void main(String[] args)
    {
        A a = new C();

        a.myMethod();
    }
}
```

- ○ My Method
- ◉ Complier error
- ○ No output

Correct answer

- ◉ My Method

) Does below code compile successfully? If not, why?

```
interface A
{
    int i = 111;
}

class B implements A
{
    void methodB()
    {
        i = 222;
    }
}
```

- ( ) Yes
- (•) No ✓

✓ 1/1

) What will be the output of the following program?

```
interface P
{
    String p = "PPPP";

    String methodP();
}

interface Q extends P
{
    String q = "QQQQ";

    String methodQ();
}

class R implements P, Q
{
    public String methodP()
    {
        return q+p;
    }

    public String methodQ()
    {
        return p+q;
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        R r = new R();

        System.out.println(r.methodP());

        System.out.println(r.methodQ());
    }
}
```

QQQQPPPP PPPPQQQQ ✓

PPPPQQQQ QQQQPPPP

✓ 1/1

```
1  interface X
2  {
3      char c = 'A';
4
5      char methodX();
6  }
7
8  class Y implements X
9  {
10     {
11         System.out.println(c);
12     }
13
14     public char methodX()
15     {
16         char c = this.c;
17
18         return ++c;
19     }
20 }
21
22 public class MainClass
23 {
24     public static void main(String[] args)
25     {
26         Y y = new Y();
27
28         System.out.println(y.methodX());
29
30         System.out.println(y.c);
31
32         System.out.println(X.c);
33     }
34 }
```

- (●) A B A A ✓
- ( ) B A B A
- ( ) A A B A

✓ 1/1

Below class ABC doesn't have even a single abstract method, but it has been declared as abstract. Is it ırrect?

```
abstract class ABC
{
    void firstMethod()
    {
        System.out.println("First Method");
    }

    void secondMethod()
    {
        System.out.println("Second Method");
    }
}
```

- (●) yes ✓
- ( ) no

✗ Can we declare protected methods in an interface? 0/1

- (●) Yes ✗
- ( ) No
- ( ) Maybe

Correct answer

- (●) No

✓ Can a class implement more than one interfaces? 1/1

- (●) Yes ✓
- ( ) No
- ( ) Maybe

✗ For every interface written in a java file, .class file will be generated after compilation? True or False?Untitled Question 0/1

- ○ True
- ◉ False ✗

Correct answer

- ◉ True

✓ 1/1

What will be the output of the below program?

```
abstract class X
{
    public X()
    {
        System.out.println("ONE");
    }

    abstract void abstractMethod();
}

class Y extends X
{
    public Y()
    {
        System.out.println("TWO");
    }

    @Override
    void abstractMethod()
    {
        System.out.println("THREE");
    }
}

public class MainClass
{
    public static void main(String[] args)
    {
        X x = new Y();

        x.abstractMethod();
    }
}
```

- (●) One two three ✓
- ( ) three two one

✗ ···/1

Why the below code is showing compile time error?

```
1  interface X
2  {
3      void methodX();
4  }
5
6  class Y implements X
7  {
8      void methodX()
9      {
10         System.out.println("Method X");
11     }
12 }
```

necaouse public access specifier not mention in derived class ✗

Correct answer

public

Feedback

Interface methods must be implemented as public. Because, interface methods are public by default and you should not reduce the visibility of any methods while overriding.

✗ ···/1

) Why the below class is showing compilation error?

```
abstract class AbstractClass
{
    abstract void abstractMethod()
    {
        System.out.println("First Method");
    }
}
```

abstract method doesn't need any defination and body ✗

Correct answer

abstract methods must not have a body

✗ ···/1

) Is the following program written correctly? If yes, what value "result" variable will hold if you run the rogram?

```
abstract class Calculate
{
    abstract int add(int a, int b);
}

public class MainClass
{
    public static void main(String[] args)
    {
        int result = new Calculate()
        {
            @Override
            int add(int a, int b)
            {
                return a+b;
            }
        }.add(11010, 022011);
    }
}
```

error-because we can't instatiated abstract method ✗

Correct answer

20235

✗ ···/1

Can you identify the error in the below code?

```
1  abstract class AbstractClass
2  {
3      private abstract int abstractMethod();
4  }
```

abstract method return type is only void it doesn't return anything ✗

Correct answer

abstract methods can’t be private.

✗ How do you print the value of field ‘i’ of interface ‘OneTwoThree’ in the below example and what will be the it’s value? ···/1

```
1  interface One
2  {
3      int i = 222;
4
5      interface OneTwo
6      {
7          int i = One.i+One.i;
8
9          interface OneTwoThree
10         {
11             int i = OneTwo.i + OneTwo.i;
12         }
13     }
14 }
```

222222222222 ✗

Correct answer

888

✗ Can we declare an interface as ‘abstract’? 0/1

- ○ Yes
- ◉ No ✗
- ○ Maybe

Correct answer

- ◉ Yes

✗ ···/1

) Can you identify the error in the below code?

```
1  interface A
2  {
3      void methodA();
4  }
5
6  class B implements A
7  {
8      public void methodA()
9      {
0          interface C
1          {
2              int i = 123;
3          }
4      }
5  }
```

we can't declare interface inside class ✗

✗ Can interfaces have static methods? 0/1

- ○ Yes
- ◉ No ✗
- ○ Maybe

Correct answer

- ◉ Yes

✓ 1/1

```
1  class A implements B
2  {
3      public int methodB(int i)
4      {
5          return i =+ i * i;
6      }
7  }
8
9  interface B
10 {
11     int methodB(int i);
12 }
13
14 public class MainClass
15 {
16     public static void main(String[] args)
17     {
18         B b = new A();
19
20         System.out.println(b.methodB(2));
21     }
22 }
```

- (●) 4 ✓
- ( ) 3
- ( ) 5
- ( ) 1

✗ ···/1

Can you identify the error in the below code?

```
interface A
{
    private int i;
}
```

✗

Correct answer

Illegal Modifier

Feedback

Only public, static and final are allowed.

Student_Id *

200240320094

✓ 1/1

Which class is instantiable? Class A or Class B?

```
1  abstract class A
2  {
3
4  }
5
6  class B extends A
7  {
8
9  }
```

- ○ Class A
- ◉ Class B ✓

✓ All members of interface are public by default. True or false? 1/1

- ◉ True ✓
- ○ False

✓ 1/1

```
interface ABC
{
    public void methodOne();

    public void methodTwo();
}

interface PQR extends ABC
{
    public void methodOne();

    public void methodTwo();
}
```

- ○ Compiler Error
- ○ Run time error
- ◉ No error ✓

✓ Can interfaces have constructors? 1/1

- ○ yes
- ◉ no ✓

✗ Like classes in java, Interfaces also extend java.lang.Object class by default. True OR False? 0/1

- (●) True ✗
- ( ) False

Correct answer

- (●) False

✓ 1/1

Is the following code written correctly?

```
1  class A
2  {
3      //Class A
4  }
5
6  interface B extends A
7  {
8      //Interface B extending Class A
9  }
```

- (●) No ✓
- ( ) Yes

✗ ···/1

Below code snippet is showing compilation error? Can you suggest the corrections?

```
1  abstract class A
2  {
3      abstract int add(int a, int b);
4  }
5
6  class B extends A
7  {
8
9  }
```

derived class not given implementation of the base class ✗

···/1

) What will be the output of the below program?

```
interface X
{
    void method();
}

class Y
{
    public void method()
    {
        System.out.println("CLASS Y");
    }
}

class Z extends Y implements X
{

}

public class MainClass
{
    public static void main(String[] args)
    {
        X x = new Z();

        x.method();
    }
}
```

error

Correct answer

Class Y