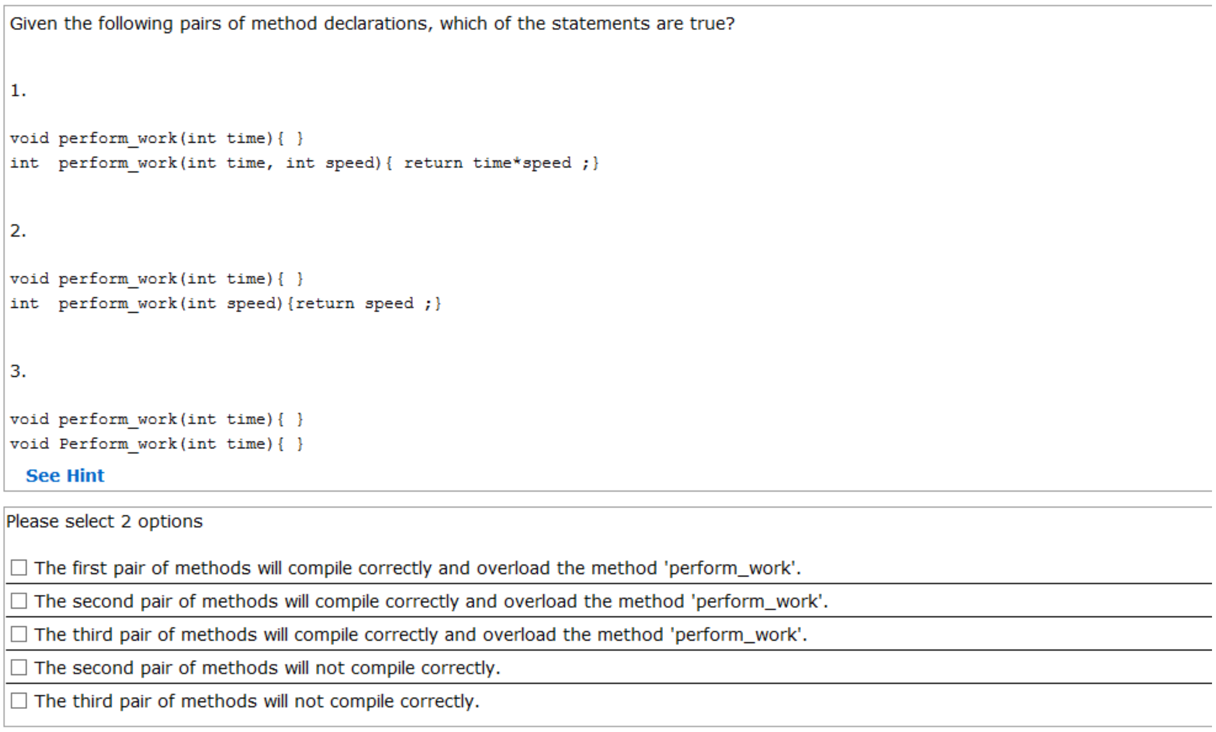
1.


Given the following pairs of method declarations, which of the statements are true?

1.

```
void perform_work(int time){ }
int  perform_work(int time, int speed){ return time*speed ;}
```

2.

```
void perform_work(int time){ }
int  perform_work(int speed){return speed ;}
```

3.

```
void perform_work(int time){ }
void Perform_work(int time){ }
```

See Hint

Please select 2 options

☐ The first pair of methods will compile correctly and overload the method 'perform_work'.

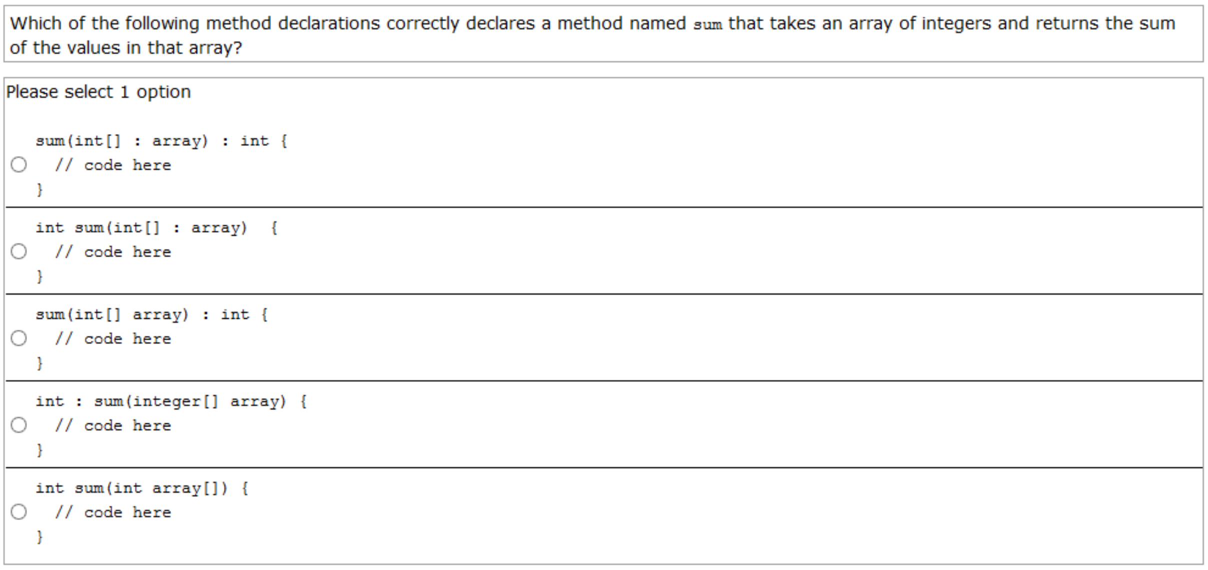
☐ The second pair of methods will compile correctly and overload the method 'perform_work'.

☐ The third pair of methods will compile correctly and overload the method 'perform_work'.

☐ The second pair of methods will not compile correctly.

☐ The third pair of methods will not compile correctly.

2.


Which of the following method declarations correctly declares a method named `sum` that takes an array of integers and returns the sum of the values in that array?

Please select 1 option

○
```
sum(int[] : array) : int {
  // code here
}
```

○
```
int sum(int[] : array)  {
  // code here
}
```

○
```
sum(int[] array) : int {
  // code here
}
```

○
```
int : sum(integer[] array) {
  // code here
}
```

○
```
int sum(int array[]) {
  // code here
}
```

3.

What would be the result of attempting to compile and run the following program?

```
class TestClass
{
   static TestClass ref;
   String[] arguments;
   public static void main(String args[])
   {
      ref = new TestClass();
      ref.func(args);
   }
   public void func(String[] args)
   {
      ref.arguments = args;
   }
}
```

Please select 1 option

- ○ The program will fail to compile, since the static method main is trying to call the non-static method func.
- ○ The program will fail to compile, since the non-static method func cannot access the static member variable ref.
- ○ The program will fail to compile, since the argument args passed to the static method main cannot be passed on to the non-static method func.
- ○ The program will fail to compile, since method func is trying to assign to the non-static member variable 'arguments' through the static member variable ref.
- ○ The program will compile and run successfully.

4.

Consider the following method...

```
public int setVar(int a, int b, float c) {
   //valid code not shown
}
```

Which of the following methods correctly overload the above method ?

Please select 2 options

☐
```
public int setVar(int a, float b, int c)
{
  return setVar(a, c, b);
}
```

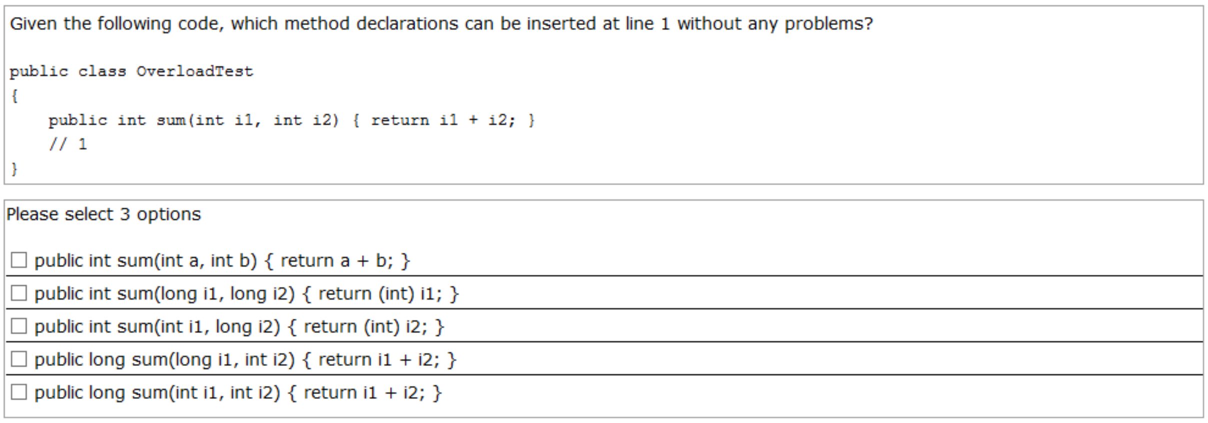
☐
```
public int setVar(int a, float b, int c)
{
  return this(a, c, b);
}
```

☐
```
public int setVar(int x, int y, float z)
{
  return x+y;
}
```

☐
```
public float setVar(int a, int b, float c)
{
  return c*a;
}
```

☐
```
public float setVar(int a)
{
  return a;
}
```

5.

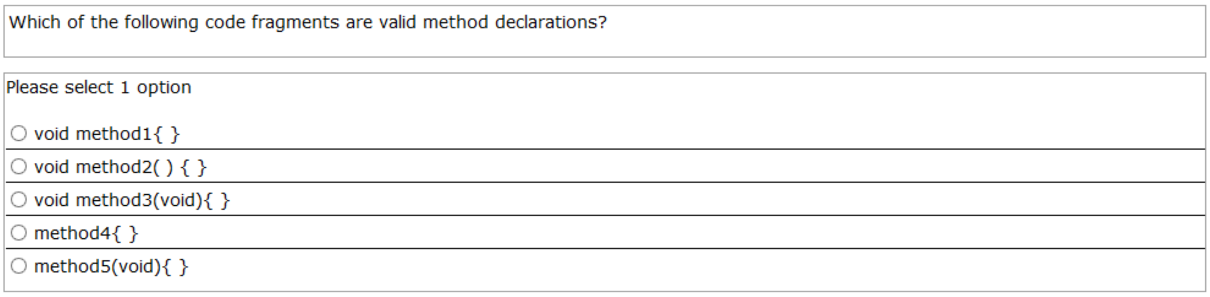

Given the following code, which method declarations can be inserted at line 1 without any problems?

```
public class OverloadTest
{
    public int sum(int i1, int i2) { return i1 + i2; }
    // 1
}
```

Please select 3 options

- ☐ public int sum(int a, int b) { return a + b; }
- ☐ public int sum(long i1, long i2) { return (int) i1; }
- ☐ public int sum(int i1, long i2) { return (int) i2; }
- ☐ public long sum(long i1, int i2) { return i1 + i2; }
- ☐ public long sum(int i1, int i2) { return i1 + i2; }

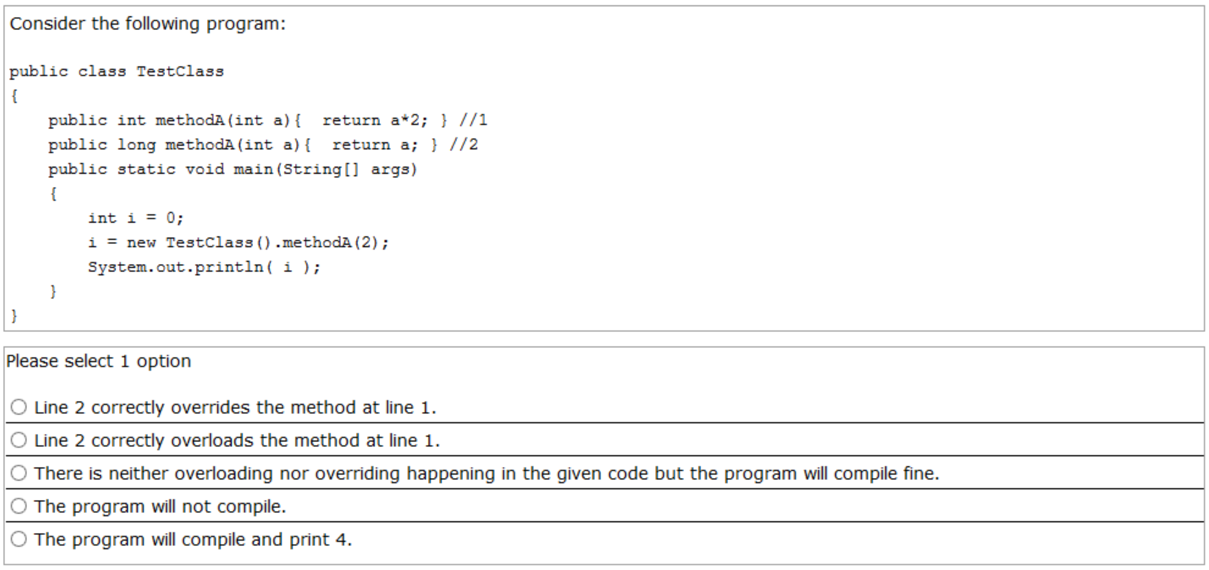
6.


Which of the following code fragments are valid method declarations?

Please select 1 option

- ○ void method1{ }
- ○ void method2( ) { }
- ○ void method3(void){ }
- ○ method4{ }
- ○ method5(void){ }

7.


Consider the following program:

```
public class TestClass
{
    public int methodA(int a){  return a*2; } //1
    public long methodA(int a){  return a; } //2
    public static void main(String[] args)
    {
        int i = 0;
        i = new TestClass().methodA(2);
        System.out.println( i );
    }
}
```

Please select 1 option

- ○ Line 2 correctly overrides the method at line 1.
- ○ Line 2 correctly overloads the method at line 1.
- ○ There is neither overloading nor overriding happening in the given code but the program will compile fine.
- ○ The program will not compile.
- ○ The program will compile and print 4.

8.

Given:

```
class Node{
    static final int TYPE = 100;
    public static void print(){
        System.out.println(TYPE); //1
    }
}

public class Test{
    public static void main(String[] args) {
       //INSERT CODE HERE //2
    }
}
```

What may be done to the above code to make it print 100?

Please select 1 option

- ○ Change the statement at //1 to `System.out.println(Node.TYPE);` and insert `Node.print();` at //2
- ○ insert `new Node().print();` at //2
- ○ insert `new Node.print();` at //2
- ○ insert `Node().print();` at //2
- ○ insert `print();` at //2.

9.

Consider the following class definition:

```
public class TestClass
{
   public static void main(){  new TestClass().sayHello(); }   //1
   public static void sayHello(){ System.out.println("Static Hello World"); }  //2
   public void sayHello() { System.out.println("Hello World "); }  //3
}
```

What will be the result of compiling and running the class?

Please select 1 option

- ○ It will print 'Hello World'.
- ○ It will print 'Static Hello World'.
- ○ Compilation error at line 2.
- ○ Compilation error at line 3.
- ○ Runtime Error.

10.

What will the following code print when run?

```
public class Mambo {

public static String makeItBetter(String str) {
  return str+"!!!";
}

public static void main(String args[]){
   String str = "Hi";
   str = makeItBetter(str);
   System.out.println(str);

}
}
```

Please select 1 option

- ○ Hi!!!
- ○ Hi
- ○ Hi!!!!!!
- ○ None of these.

11.

Given:

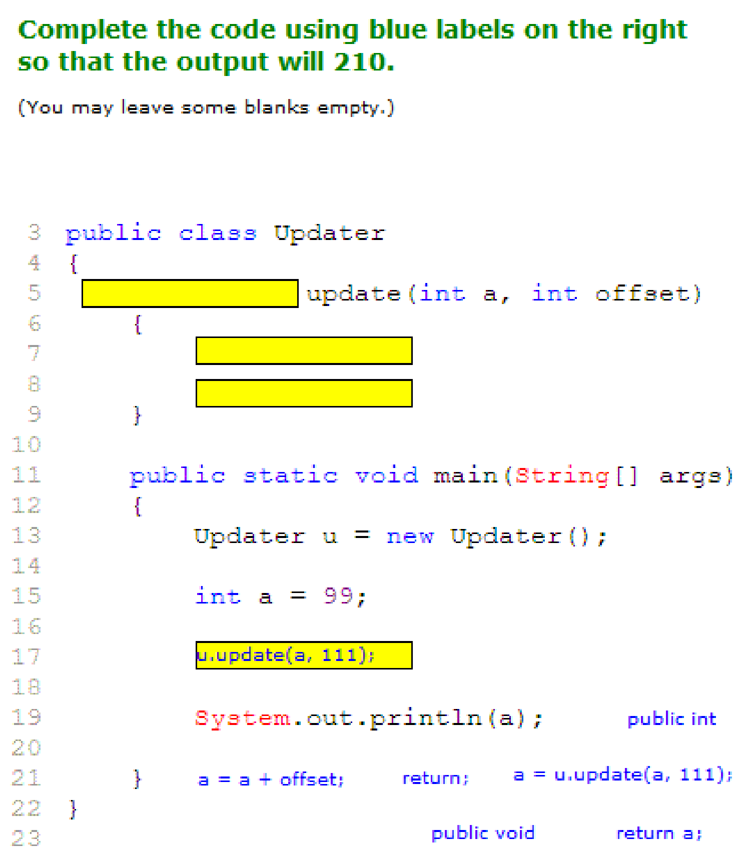
```
public class Test
{
    static int a;
    int b;

    public void incr(){
       int c = a++;
       b++;
       c++;
       System.out.println(a+" "+b+" "+c);
    }
   public static void main(String args[])
   {
      Test test = new Test();
      test.incr();
      a++;
      test = new Test();
      test.incr();
   }
}
```

What will be the output?

Please select 1 option

- ○ Compilation failure.
- ○ 1 1 1
  2 1 2
- ○ 1 1 1
  3 1 3
- ○ 1 2 1
  2 3 3
- ○ 1 2 1
  3 3 3

12.



Complete the code using blue labels on the right so that the output will 210.

(You may leave some blanks empty.)

```
public class Updater
{
                   update(int a, int offset)
    {


    }

    public static void main(String[] args)
    {
        Updater u = new Updater();

        int a = 99;

        u.update(a, 111);

        System.out.println(a);

    }
}
```

```
public int
a = a + offset;
return;
a = u.update(a, 111);
public void
return a;
```

13.

Which of the following statements are true?

Please select 2 options

- ☐ A static method can call other non-static methods in the same class by using the 'this' keyword.
- ☐ A class may contain both static and non-static variables and both static and non-static methods.
- ☐ Every object of a class has its own instance of each non-static member variable.
- ☐ Instance methods may access local variables of static methods.
- ☐ All methods in a class are implicitly passed a 'this' parameter when called.

14.

Consider the following code:

```
public class Varargs
{
  public void test()
  {
      test1(10);      //1
      test1(10, 20); //2
  }

  public static void main(String[] args)
  {
    new Varargs().test();
  }

  //insert method here.
}
```

Which of the following lines can be added independently to the above class so that it will run without any errors or exceptions?

Please select 2 options

☐ public void test1(int i, int j){}

☐ public void test1(int i, int... j){}

☐ public void test1(int... i){}

☐ public void test1(int i...){}

☐ public void test1(int[] i){}

15.

What will be the result of attempting to compile and run the following class?

```
public class TestClass
{
    public static void main(String args[ ] )
    {
       int i = 1;
       int[] iArr = {1};
       incr(i) ;
       incr(iArr) ;
       System.out.println( "i = " + i + "  iArr[0] = " + iArr [ 0 ] ) ;
    }

    public static void incr(int   n ) { n++ ; }

    public static void incr(int[ ] n ) { n [0]++ ; }
}
```

See Hint

Please select 1 option

○ The code will print `i = 1 iArr[0] = 1`

○ The code will print `i = 1 iArr[0] = 2`

○ The code will print `i = 2 iArr[0] = 1`

○ The code will print `i = 2 iArr[0] = 2`

○ The code will not compile.

16.

What will the following class print when compiled and run?

```
public class Holder
{
   int value = 1;
   Holder link;
   public Holder(int val){ this.value = val; }
   public static void main(String[] args)
   {
        Holder a = new Holder(5);
        Holder b = new Holder(10);
        a.link = b;
        setIt(a, b);
        System.out.println(a.link.value+", "+b.link.value);
   }

   public static void setIt(Holder x, Holder y)
   {
       y.link = x;
   }

}
```

Please select 1 option

- ○ It will print 5, 5.
- ○ It will print 10, 5.
- ○ It will print 5, 10.
- ○ It will print 10, 10.
- ○ None of these.

17.

How can you declare a method someMethod() such that an instance of the class is not needed to access it and all the members of the same package have access to it.

Please select 3 options

- ☐ public static void someMethod()
- ☐ static void someMethod()
- ☐ protected static void someMethod()
- ☐ void someMethod()
- ☐ protected void someMethod()
- ☐ public abstract static void someMethod()