Ex1

Given:

```
class Overloading {
        int x(double d) {
            System.out.println("one");
            return 0;
       }
        String x(double d) {
            System.out.println("two");
            return null;
       }
        double x(double d) {
            System.out.println("three");
            return 0.0;
       }
  }

class Test
{
       public static void main(String[] args) {
                    Overloadint o=new Overloading();
                    o.x(4.0);
              }

}
```

What is the result?

A.One

B.Two

C.Three

D.Compilation fails.

Ex2

Given the following class:

```
public class CheckingAccount {
        public int amount;
        public CheckingAccount(int amount) {
            this.amount = amount;
       }
        public int getAmount() {
            return amount;
       }
```

```
    public void changeAmount(int x) {
        amount += x;
    }
}
```

And given the following main method, located in another class:

```
public static void main(String[] args){
    CheckingAccount acct = new CheckingAccount((int)(Math.random() * 1000));
    //line n1
    System.out.println(acct.getAmount());
}
```

Which three lines, when inserted independently at line n1, cause the program to print a 0 balance?

A.this.amount = 0;
B.amount = 0;
C.acct(0);
D.acct.amount = 0;
E.acct.getAmount() = 0;
F.acct.changeAmount(0);
G.acct.changeAmount(-acct.amount);
H.acct.changeAmount(-acct.getAmount());

Ex3
Given the code fragment:

```
public static void main(String[] args) {
    int ii = 0;
    int jj = 7;
    for (ii = 0; ii < jj - 1; ii = ii + 2) {
        System.out.print(ii + " ");
    }
}
```

What is the result?

A.2 4
B.0 2 4 6
C.0 2 4
D.Compilation fails

Ex4
Given:

```
int i, j = 0;
```

```
i = (3 * 2 + 4 + 5);
j = (3 * ((2 + 4) + 5));
System.out.println("i:" + i + "\nj:" + j);
```

What is the result?

A.
i:16
j:33

B.
i.15
j:33

C.
i:33
j:23

D.
i:15
j:23

Ex5
Given the code fragment:

```
float x = 22.00f % 3.00f;
int y = 22 % 3;
System.out.print(x + ", " + y);
```

What is the result?

A.1.0, 1
B.1.0f, 1
C.7.33, 7
D.Compilation fails
E.An exception is thrown at runtime

Ex6
Given the code fragment:

```
Int[] a =new int[] {1, 2, 3, 4, 5};
for (XXX) {
    System.out.print(a[e]);
}
```

Which option can replace xxx to enable the code to print 135?

A.int e = 0; e <= 4; e++
B.int e = 0; e < 5; e += 2
C.int e = 1; e <= 5; e += 1
D.int e = 1; e < 5; e +=2

Ex7
What is the proper way to defined a method that take two int values and returns their sum as an int value?

A.int sum(int first, int second) { first + second; }
B.int sum(int first, second) { return first + second; }
C.sum(int first, int second) { return first + second; }
D.int sum(int first, int second) { return first + second; }
E.void sum (int first, int second) { return first + second; }

Ex8
Given:

```
public class App {
        public static void main(String[] args) {
            Boolean[] bool = new Boolean[2];
            bool[0] = new Boolean(Boolean.parseBoolean("true"));
            bool[1] = new Boolean(null);
            System.out.println(bool[0] + " " + bool[1]);
      }
}
```

What is the result?

A.true false
B.true null
C.Compilation fails
D.A NullPointerException is thrown at runtime

Ex9
Given:

```
public class TestLoop {

      public static void main(String[] args) {
            int[] array =new int[] {0, 1, 2, 3, 4};
```

```
        int key = 3;
        for (int pos = 0; pos < array.length; ++pos) {
            if (array[pos] == key) {
                break;
            }
        }
        System.out.print("Found " + key + " at " + pos);
    }
}
```

What is the result?

A. Found 3 at 2
B. Found 3 at 3
C. Compilation fails
D. An exception is thrown at runtime

Ex10

Given the following array:

```
int[] intArr =new int[] {8, 16, 32, 64, 128};
```

Which two code fragments, independently, print each element in this array?

A.

```
for (int i : intArr) {
    System.out.print(intArr[i] + " ");
}
```

B.

```
for (int i : intArr) {
    System.out.print(i + " ");
}
```

C.

```
for (int i = 0; intArr) {
    System.out.print(intArr[i] + " ");
    i++;
}
```

D.

```
for (int i = 0; i < intArr.length; i++) {
	System.out.print(i + " ");
}
```

E.

```
for (int i = 0; i < intArr.length; i++) {
	System.out.print(intArr[i] + " ");
}
```