A+ Computer Science

AP REVIEW

2017 AP CS A EXAM

Provided by A+ Computer Science

Visit us at

www.apluscompsci.com

Full Curriculum Solutions

M/C Review Question Banks

Live Programming Problems

Tons of great content!

www.facebook.com/APlusComputerScience

Multiple Choice

-answer the easiest question 1st

-work through the test more than once

-use the test to take the test

-work more time intensive problems last

-bubble answers on answer sheet as you go

-answer every question

-keep track of your time - 90 minutes

Free Response

-Read all 4 questions before writing anything

-answer the easiest question 1st

-most times question 1 is the easiest

-see if part B calls part A and so on

-many times part C consists of A and B calls

-write something on every question

-write legibly / use PENCIL!!!!!!!!!!

-keep track of your time

Free Response

-When writing methods

  -use parameter types and names as provided

  -do not redefine the parameters listed

  -do not redefine the methods provided

  -return from all return methods

  -return correct data type from return methods

Free Response

-When writing a class or methods for a class

  -know which methods you have

  -know which instance variables you have

  -check for public/private on methods/variables

  -return from all return methods

  -return correct data type from return methods

Free Response

-When extending a class

  -know which methods the parent contains

  -have the original class where you can see it

  -make sure you have super calls

  -check for public/private on methods/variables

  -make super calls in sub class methods as needed

Free Response

-When extending abstract / implementing interface

  -know which methods the parent contains

  -have the original class where you can see it

  -make sure you have super calls

  -check for public/private on methods/variables

  -make super calls in sub class methods as needed

  -implement all abstract methods in sub class

Free Response Topics

ArrayList

– get,set,remove,add,size

Make a Class – Implement an Interface

– create a class that implements an interface

Strings

– Fun with Strings - indexOf, substring

Array or Arrays of References

– nested loops - array of arrays concepts

Provided by A+ Computer Science

Visit us at

www.apluscompsci.com

Full Curriculum Solutions

M/C Review Question Banks

Live Programming Problems

Tons of great content!

www.facebook.com/APlusComputerScience

Free Response Question 1

ArrayList % mod

Remainders

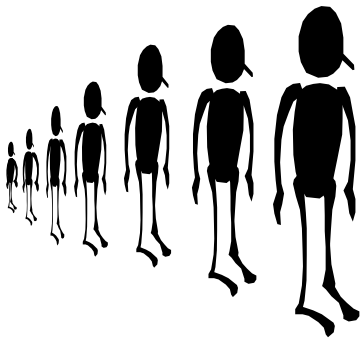
```
int num = 193054;
out.println( num % 10 );

num = num / 10;
out.println( num % 10 );

num = num / 10;
out.println( num % 10 );

num = num / 10;
out.println( num % 10 );
```

OUTPUT

```
4
5
0
3
```

ArrayList

A typical ArrayList question involves putting something into an ArrayList and removing something from an ArrayList.



ArrayList

Arraylist is a class that houses an array.

An ArrayList can store any type.

All ArrayLists store the first reference at spot / index position 0.

ArrayList

frequently used methods

Name	Use
add(item)	adds item to the end of the list
add(spot,item)	adds item at spot – shifts items up->
set(spot,item)	put item at spot z[spot]=item
get(spot)	returns the item at spot return z[spot]
size()	returns the # of items in the list
remove()	removes an item from the list
clear()	removes all items from the list

```
import  java.util.ArrayList;
```

ArrayList

```
List<String> ray;
ray = new ArrayList<String>();
ray.add("hello");
ray.add("whoot");
ray.add("contests");
out.println(ray.get(0).charAt(0));
out.println(ray.get(2).charAt(0));
```

OUTPUT

```
h
c
```

ray stores String references.

ArrayList

```
int spot=list.size()-1;
while(spot>=0)
{
   if(list.get(spot).equals("killIt"))
      list.remove(spot);

   spot--;

}
```

ArrayList

```
for(int spot=list.size()-1; i>=0; i--)
{
   if(list.get(spot).equals("killIt"))
      list.remove(spot);
}
```

ArrayList

```
int spot=0;
while(spot<list.size())
{
   if(list.get(spot).equals("killIt"))
      list.remove(spot);
   else
      spot++;
}
```

```
public Digits( int num )
{
  digitList = new ArrayList<Integer>();

  if( num == 0 )
    digitList.add( 0 );

  while( num > 0 )
  {
    digitList.add( 0, num % 10 );
    num = num / 10;
  }
}
```

2017 Question 1 Part A

```
public boolean isStrictlyIncreasing()
{
  for( int i = 0 ; i < digitList.size()-1; i++ )
  {
    if( digitList.get( i ) >= digitList.get( i+1 ) )
      return false;
  }
  return true;
}
```

2017 Question 1 Part B

Provided by A+ Computer Science

Visit us at

www.apluscompsci.com

Full Curriculum Solutions

M/C Review Question Banks

Live Programming Problems

Tons of great content!

www.facebook.com/APlusComputerScience

Free Response Question 2

Implement an Interface Make a class

Make a Class

```
public Triangle(int a, int b, int c)
{
   sideA=a;
   sideB=b;
   sideC=c;
}
```

Constructors are similar to methods. Constructors set the properties of an object to an initial state.

Make a Class

```
public void setSideA(int a )
{
    sideA=a;
}
```

Modifier methods are methods that change the properties of an object.

Make a Class

```
public int getSideA()
{
  return sideA;
}
```

Accessor methods are methods that retrieve or grant access to the properties of an object, but do not make any changes.

Make a Class

```
public class Triangle
{
   private int sideA;
   private int sideB;
   private int sideC;
```

Instance variables store the state information for an object.

Abstract / Interfaces

```
public interface Exampleable
{
   int writeIt(Object o);
   int x = 123;
}
```

Methods are public abstract!
Variables are public static final!

Abstract / Interfaces

```
public interface Exampleable
{
   public abstract int writeIt(Object o);
   public static final int x = 123;
}
```

Methods are public abstract!
Variables are public static final!

Abstract / Interfaces

An interface is a list of abstract methods that must be implemented.

An interface may not contain any implemented methods.

Interfaces cannot have constructors!!!

Abstract / Interfaces

Interfaces are typically used when you know what you want an Object to do, but do not know how it will be done.

If only the behavior is known, use an interface.

```
public class MultPractice implements StudyPractice
{
  private int first;
  private int second;

  public MultPractice( int f, int s )
  {
    first = f;
    second = s;
  }

  public String getProblem()
  {
   return "" + first + " TIMES " + second;
  }

  public void nextProblem()
  {
   second++;
  }

  public String toString()
  {
    return "" + first + " " + second;
  }
}
```

Make a Class

2017 Question 2

Provided by A+ Computer Science

Visit us at

www.apluscompsci.com

Full Curriculum Solutions

M/C Review Question Banks

Live Programming Problems

Tons of great content!

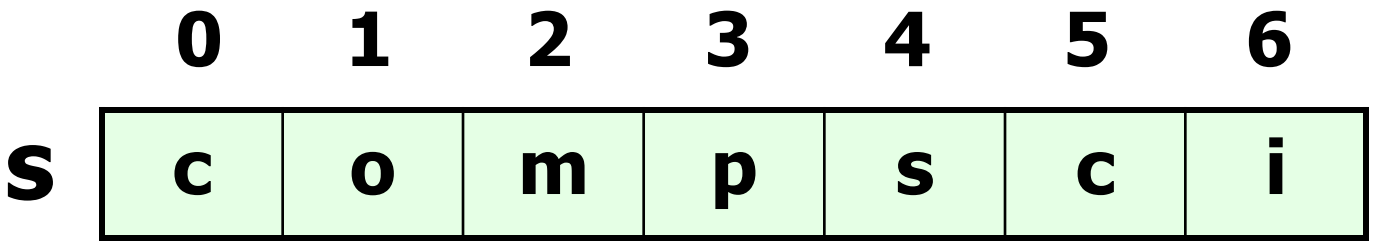
www.facebook.com/APlusComputerScience

Free Response Question 3

String Love Part II

String Love

String s = "compsci";



A string is a group of characters.
The first character in the group is at spot 0.

String

frequently used methods

Name	Use
substring(x,y)	returns a section of the string from x to y not including y
substring(x)	returns a section of the string from x to length-1
length()	returns the # of chars
charAt(x)	returns the char at spot x
indexOf(c)	returns the loc of char c in the string, searching from spot 0 to spot length-1
lastIndexOf(c)	returns the loc of char c in the string, searching from spot length-1 to spot 0

String

frequently used methods

Name	Use
equals(s)	checks if this string has same chars as s
compareTo(s)	compares this string and s for >,<, and ==
trim()	removes leading and trailing whitespace
replaceAll(x,y)	returns a new String with all x changed to y
toUpperCase()	returns a new String with uppercase chars
toLowerCase()	returns a new String with lowercase chars

String Love

```
String sent = "alligators rule";
String find = "gato";

System.out.println( sent.indexOf( find ) );
System.out.println( sent.indexOf( "dog" ) );
System.out.println( sent.substring( 3 , 6 ) );
System.out.println( sent.substring( 6 ) );
```

OUTPUT

```
4
-1
iga
tors rule
```

2017 Question 3 part A

```
int pos = findNthOccurence( str, n);

if( pos != -1 )
{
  String f = currentPhrase.substring( 0 , pos);
  String s = currentPhrase.substring( pos+str.length() );
  currentPhrase = f + repl + s;
}
```

2017 Question 3 - part B

```
//probably the preferred way
int cnt = 1;
int st = findNthOccurence( str, cnt );
while( st != -1 )
{
  cnt++;
  int chk  = findNthOccurence( str, cnt );
  if( chk == -1 )
    return st;
  st = chk;
}
return st;
```

2017 Question 3 - part B

```
//uses the findNth method but a bit weird
int st = currentPhrase.length()-str.length();
while( st >= 0 )
{
  int pos = findNthOccurence( str, st);
  if( pos != -1 )
    return pos;
  st = st - 1;
}
return -1;

//a cool way that does not follow the rules
return currentPhrase.lastIndexOf( str );
```

Provided by A+ Computer Science

Visit us at

www.apluscompsci.com

Full Curriculum Solutions

M/C Review Question Banks

Live Programming Problems

Tons of great content!

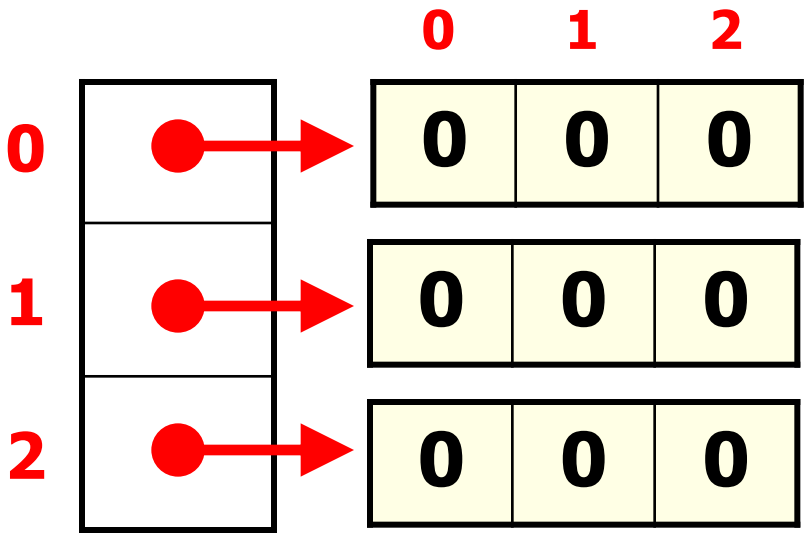
www.facebook.com/APlusComputerScience

Free Response Question 4

Matrix of References

Matrices

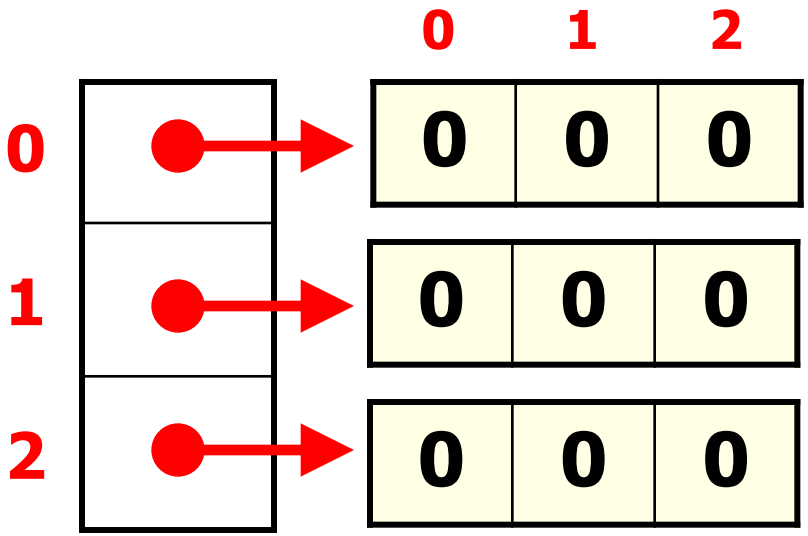
Typically, 1 question on the A test free response will require that students manipulate a 2-dimensional array.



Matrices

A matrix is an array of arrays.

```
int[][] mat = new int[3][3];
```



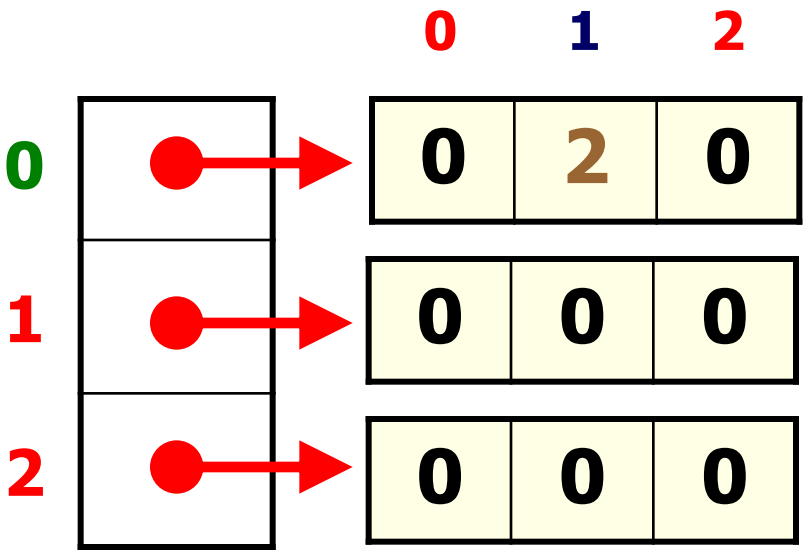
Matrices

A matrix is an array of arrays.

```
int[][] mat = new int[3][3];
mat[0][1]=2;
```

Which array?

Which spot?



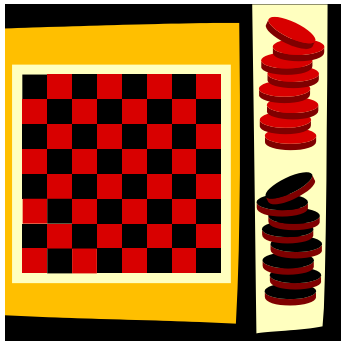
Matrices

	0	1	2	3	4
0	0	0	0	5	0
1	0	0	0	0	0
2	0	0	7	0	0
3	0	0	0	0	0
4	0	3	0	0	0

mat[2][2]=7;

mat[0][3]=5;

mat[4][1]=3



Matrices

```
for( int r = 0; r < mat.length; r++)
{
  for( int c = 0; c < mat[r].length; c++)
  {
       mat[r][c] = r*c;
  }
}
```

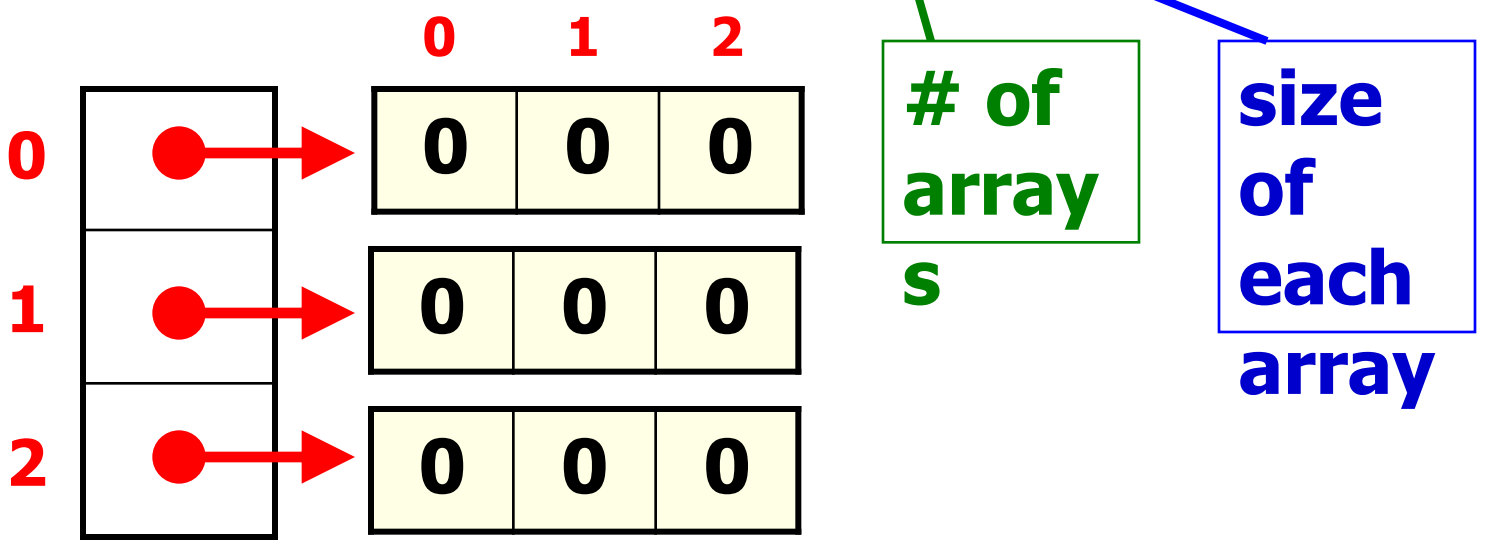
if mat was 3x3

0	0	0
0	1	2
0	2	4

Matrices

A matrix is an array of arrays.

```
int[][] mat = new int[3][3];
```



Matrices – for each

```
int[][] mat = {{5,7},{5,3,4,6},{0,8,9}};

for( int[] row : mat )
{
  for( int num : row )
  {
    System.out.print( num + " ");
  }
  System.out.println();
}
```

OUTPUT

```
5 7
5 3 4 6
0 8 9
```

Matrices – for loop

```
int[][] mat = {{5,7},{5,3,4,6},{0,8,9}};

for( int r = 0; r < mat.length; r++ )
{
  for( int c = 0; c < mat[r].length; c++ )
  {
    System.out.print( mat[r][c] + " ");
  }
  System.out.println();
}
```

OUTPUT

```
5 7
5 3 4 6
0 8 9
```

```
public static Position findPosition( int num, int[][] intArr )
{
 for( int r = 0; r < intArr.length; r++ )
 {
        for( int c = 0; c < intArr[0].length; c++ )
        {
          if( num == intArr[r][c] )
             return new Position( r, c );
        }
 }
 return null;
}
```

2017
Question 4
part A

```
public static Position[][] getSuccessorArray( int[][] intArr )
{
  Position[][] bob = new Position[intArr.length][intArr[0].length];
  for( int r = 0; r < intArr.length; r++ )
  {
     for( int c = 0; c < intArr[0].length; c++ )
     {
       bob[r][c] = findPosition( intArr[r][c]+1, intArr );
     }
  }
  return bob;
}
```

2017 Question 4 part B

Provided by A+ Computer Science

Visit us at

www.apluscompsci.com

Full Curriculum Solutions

M/C Review Question Banks

Live Programming Problems

Tons of great content!

www.facebook.com/APlusComputerScience

Multiple Choice

-answer the easiest question 1st

-work through the test more than once

-use the test to take the test

-work more time intensive problems last

-bubble answers on answer sheet as you go

-answer every question

-keep track of your time - 90 minutes

Free Response

-Read all 4 questions before writing anything

-answer the easiest question 1st

-most times question 1 is the easiest

-see if part B calls part A and so on

-many times part C consists of A and B calls

-write something on every question

-write legibly / use PENCIL!!!!!!!!!!

-keep track of your time – 90 minutes

Free Response

-When writing methods

  -use parameter types and names as provided

  -do not redefine the parameters listed

  -do not redefine the methods provided

  -return from all return methods

  -return correct data type from return methods

Free Response

-When writing a class or methods for a class

  -know which methods you have

  -know which instance variables you have

  -check for public/private on methods/variables

  -return from all return methods

  -return correct data type from return methods

Free Response

-When extending a class

  -know which methods the parent contains

  -have the original class where you can see it

  -make sure you have super calls

  -check for public/private on methods/variables

  -make super calls in sub class methods as needed

Free Response

-When extending abstract / implementing interface

  -know which methods the parent contains

  -have the original class where you can see it

  -make sure you have super calls

  -check for public/private on methods/variables

  -make super calls in sub class methods as needed

  -implement all abstract methods in sub class

Free Response Topics

ArrayList

– get,set,remove,add,size

Make a Class – Implement an Interface

– create a class that implements an interface

Strings

– Fun with Strings - indexOf, substring

Array or Arrays of References

– nested loops - array of arrays concepts

A+ Computer Science

AP REVIEW

2017 AP CS A EXAM