A+ Computer Science

AP REVIEW

2016 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

-cycle through the test more than once

-use the test to take the test

-save long tracing problems for last

-bubble 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

Make a Class - Strings / List

– create a basic class with instance variables, constructors, and methods

ArrayList of References / Strings

– get,set,remove,add,size – levels of abstraction

Array / Array or Arrays

– for and for each loops, nested loops, array of arrays concepts

List of Strings – String Love Fest III

– More List manipulation and more Strings

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

Make a Class String / List

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.

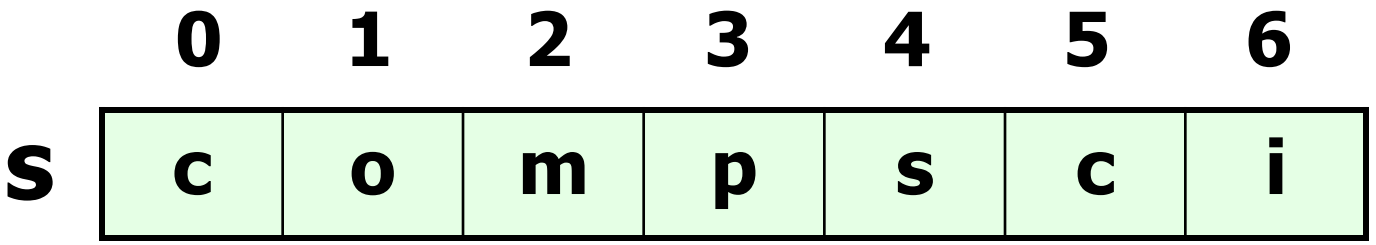
Make a Class FR Example

```
public class HiddenWord
{
  private String word;

   public HiddenWord( String w )
   {
     word = w;
   }

   public String getHint( String guess )
   {
     String ret = "";
     for( int i = 0; i < word.length(); i++)
     {
           if( word.substring(i,i+1).equals(guess.substring(i,i+1)) )
             ret += word.substring( i, i+1);
           else if( word.indexOf( guess.substring(i,i+1))!=-1)
             ret += "+";
           else
             ret += "*";
     }
     return ret;
   }

}
```

2015 Question 2

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
```

2016 Question 1 Part A

```
public class RandomStringChooser
{
   private ArrayList<String> choices;

   public RandomStringChooser( String[] list )
   {
          choices = new ArrayList<String>();
          for( String s : list )
          {
              choices.add( s );
          }
   }

   public String getNext( )
   {
          int sz = choices.size();
          if( sz == 0 )
              return "NONE";
          int spot = (int)(Math.random()*sz);
          return choices.remove( spot );
   }
}
```

```
public RandomLetterChooser( String str )
 {
   super( getSingleLetters( str ) );
 }
```

2016 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!

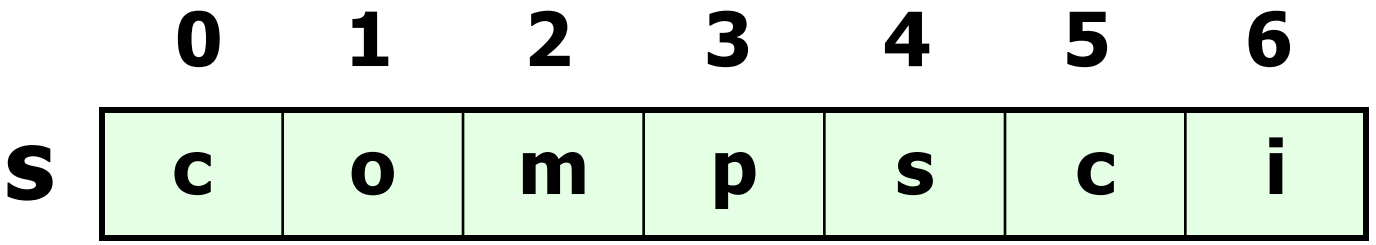
www.facebook.com/APlusComputerScience

Free Response Question 2

More Strings and Lists

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

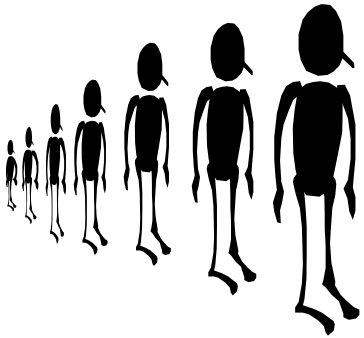
```
String sent = "alligators rule";
String find = "gato";

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
```

ArrayList

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



ArrayList

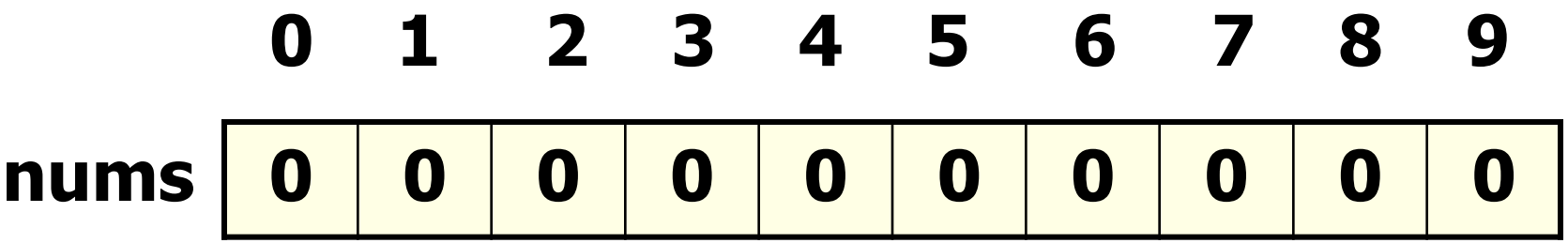
Arraylist is a class that houses an array.

An ArrayList can store any type.

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

ArrayList

```
int[] nums = new int[10];          //Java int array
```



An array is a group of items all of the same type which are accessed through a single identifier.

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++;
}
```

2016 Question 2 part A

```
public LogMessage( String message )
{
        //split would work great for this
        int spot = message.indexOf( ":" );
        machineId = message.substring( 0, spot );
        description = message.substring( spot + 1 );
  }
```

Somebody loves Strings!

2016 Question 2 - part B

```
public boolean containsWord( String keyword )
{
  //contributed by my 9th grader
  int pos1 = description.indexOf( keyword + " " );
  int pos2 = description.indexOf( " " + keyword );
  int pos3 = description.indexOf( " " + keyword + " " );
  int end = description.length() - keyword.length()-1;
  if( end+1 == 0 )
    return description.equals(keyword);
  if( pos1 == 0 ) return true;
  if( pos2 == end && end != -1)
          return true;
  if( pos3 > -1 ) return true;
  return false;
}
```

version 1

Somebody loves Strings!

2016 Question 2 - part B

```
public boolean containsWord( String keyword )
{
  String[] stuff = description.split(" ");
  for( String s : stuff )
    if( s.equals( keyword) )
      return true;
  return false;
}
```

Version 2

Somebody loves Strings!

2016 Question 2 – part C

```
public List<LogMessage> removeMessages(String keyword)
{
  ArrayList<LogMessage> a=new ArrayList<LogMessage>();
  for(int y=0;y<messageList.size();y++)
  {
    if(messageList.get(y).containsWord(keyword))
    {
      a.add(messageList.remove(y--));
    }
  }
  return a;
}
```

You must know ArrayList!

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!

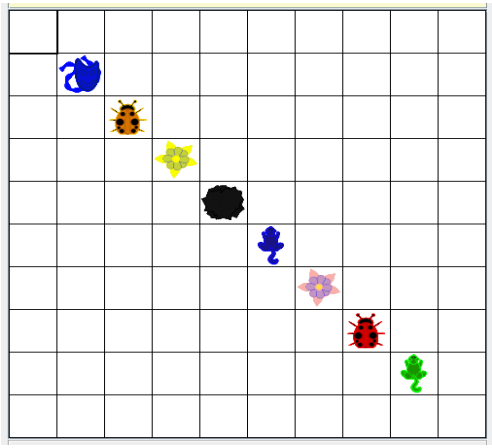
www.facebook.com/APlusComputerScience

Free Response Question 3

Matrix of References

Matrices

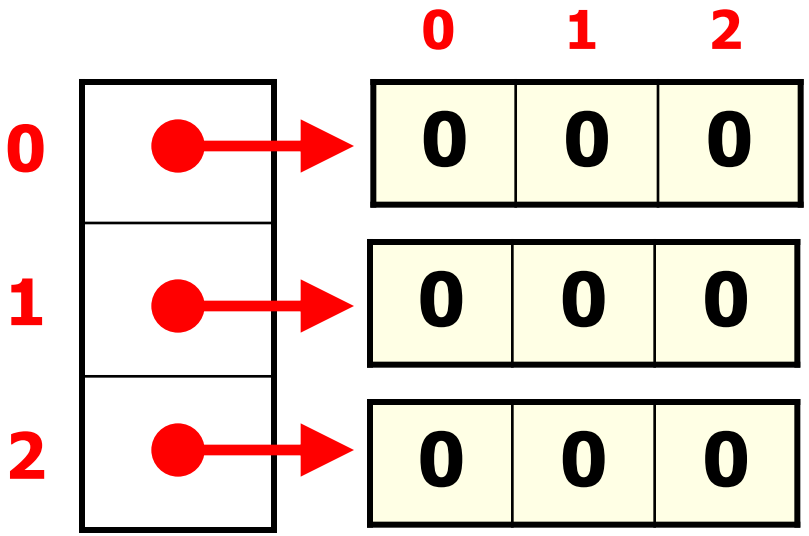
One question on the A test free response will require you to manipulate a 2-dimensional array or a GridWorld grid.



Matrices

A matrix is an array of arrays.

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



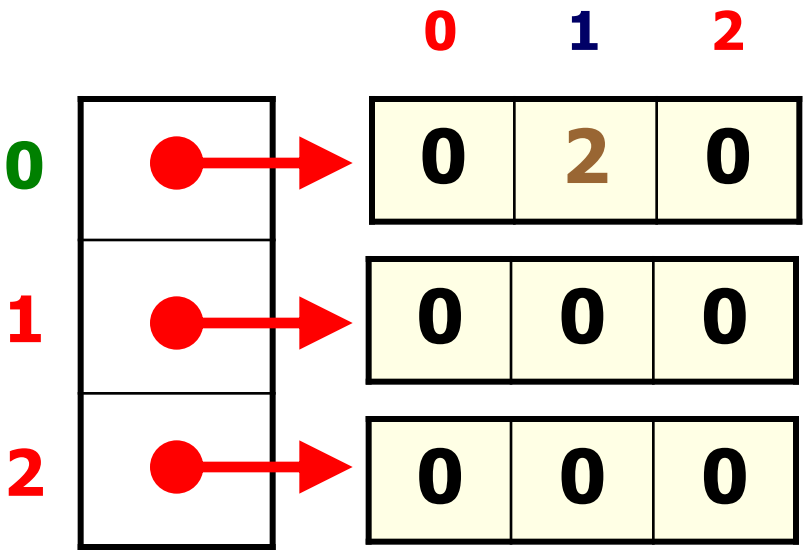
Matrices

A matrix is an array of arrays.

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

Which array?

Which spot?



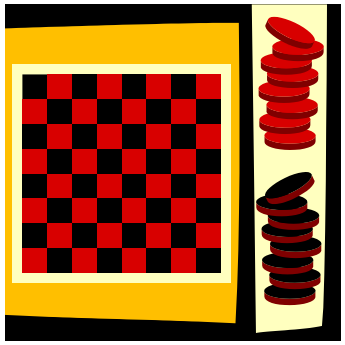
Matrices

	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;
  }
}
```

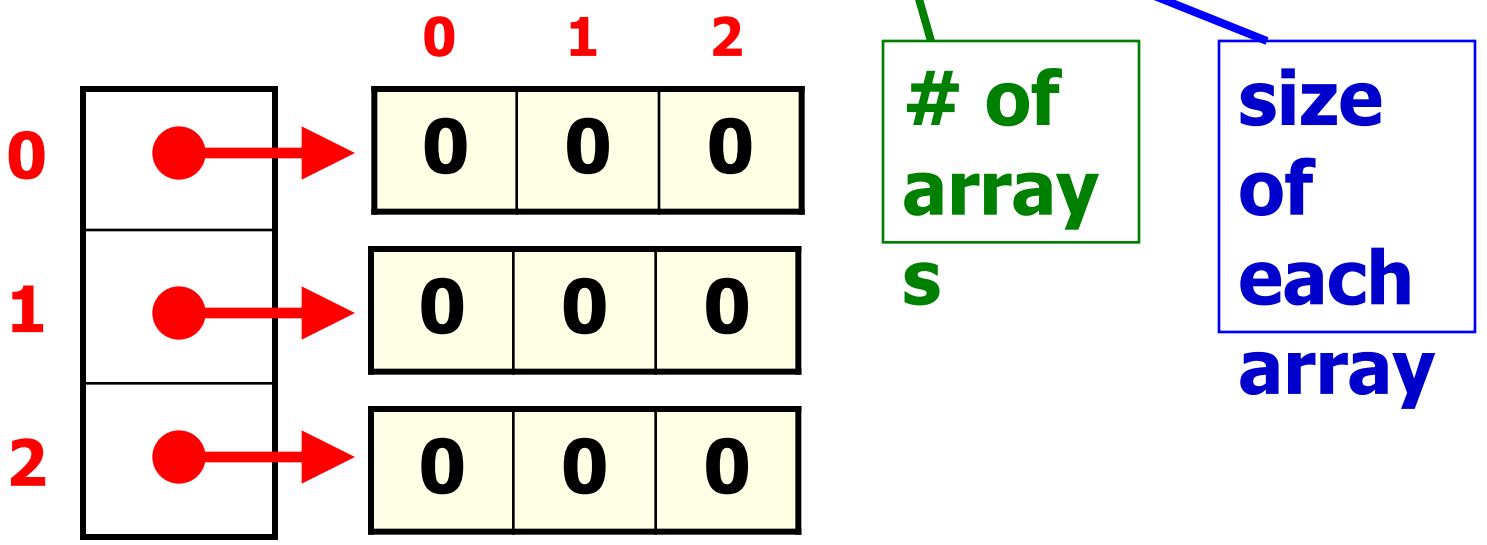
if mat was 3x3

0	0	0
0	1	2
0	2	4

Matrices

A matrix is an array of arrays.

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



Matrices

```
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
```

```
public boolean toBeLabeled( int r, int c,
                            boolean[][] blackSquares )
{
  return( ! blackSquares[r][c]  &&
       ( r==0 || c==0 ||
          blackSquares[r][c-1] ||
             blackSquares[r-1][c]) );
}
```

2016 Question 3 part A

```
public CrosswordPuzzle( boolean[][] blackSquares )
{
  int k=0;
  puzzle = new Square[ blackSquares.length ]
                                          [ blackSquares[0].length ];
  for(int r=0;r<puzzle.length;r++)
  {
        for(int c=0;c<puzzle[r].length;c++)
        {
          if(blackSquares[r][c])
                puzzle[r][c]=new Square(true,0);
          else if(toBeLabeled(r,c,blackSquares))
                puzzle[r][c]=new Square(false,++k);
          else
                puzzle[r][c]=new Square(false,0);
        }
  }
}
```

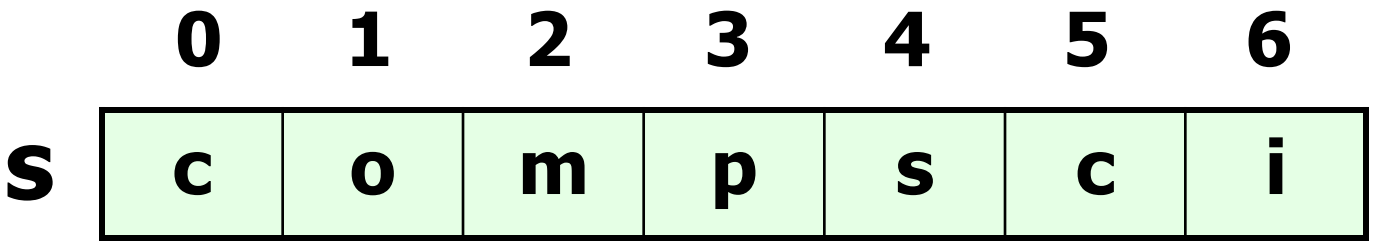
2016 Question 3 part B

Free Response Question 4

Lists and MORE Strings

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

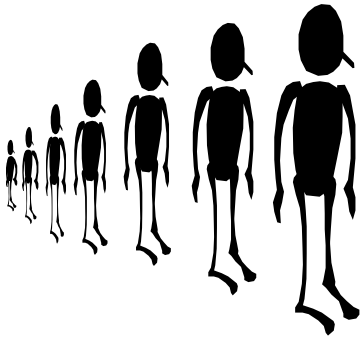
```
String sent = "alligators rule";
String find = "gato";

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
```

ArrayList

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



ArrayList

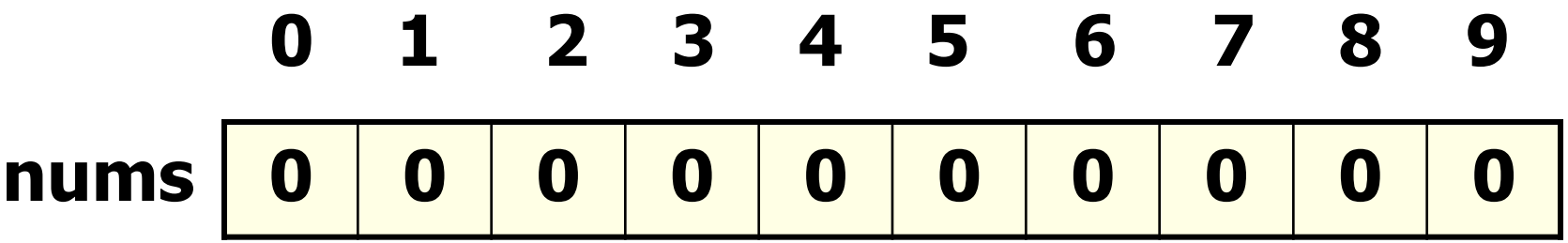
Arraylist is a class that houses an array.

An ArrayList can store any type.

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

ArrayList

```
int[] nums = new int[10];          //Java int array
```



An array is a group of items all of the same type which are accessed through a single identifier.

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 static int totalLetters( List<String> wordList )
{
  int sum = 0;
  for( String s : wordList )
  {
        sum += s.length();
  }
  return sum;
}
```

2016 Question 4 Part A

```
public static int basicGapWidth(
                List<String>  wordList, int formattedLen )
{
   int size = totalLetters( wordList );
   int diff = formattedLen - size;
   return diff / (wordList.size()-1);
}
```

2016 Question 4 Part B

```
public static String format( List<String> wordList, int formattedLen )
{
  String form = "";
  int left = leftoverSpaces( wordList, formattedLen);
  int wid =  basicGapWidth( wordList, formattedLen);
  for(int i = 0; i < wordList.size()-1; i++)
  {
    form += wordList.get(i);
    for( int j = 0; j < wid; j++)
    {
      form += " ";
    }
    if( i < left )
    {
       form +=  " ";
    }
  }
  form += wordList.get( wordList.size()-1 );
  return form;
}
```

2016
Question 4
Part C
version 1

```
public static String format( List<String> wordList, int formattedLen )
{
  ArrayList<String> cool;
  cool = new ArrayList<String>( wordList );
  String form = "";
  String spaces = "";
  for( int j = 0; j < basicGapWidth( cool, formattedLen); j++)
  {
    spaces += " ";
  }
  for(int i = 0; i < cool.size()-1; i++)
  {
    cool.set( i, cool.get(i) + spaces );
  }
  int left = leftoverSpaces( cool, formattedLen);
  for(int i = 0; i < left && i < cool.size(); i++)
  {
    cool.set( i, cool.get(i)+" " );
  }
  for( String s : cool )
  {
    form += s;
  }
  return form;
}
```

2016 Question 4 Part C version 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

Multiple Choice

-answer the easiest question 1st

-cycle through the test more than once

-use the test to take the test

-save long tracing problems for last

-bubble 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

Make a Class - Strings / List

– create a basic class with instance variables, constructors, and methods

ArrayList of References / Strings

– get,set,remove,add,size – levels of abstraction

Array / Array or Arrays

– for and for each loops, nested loops, array of arrays concepts

List of Strings – String Love Fest III

– More List manipulation and more Strings

A+ Computer Science

AP REVIEW

2016 AP CS A EXAM