A+ Computer Science

AP REVIEW

2011 FR QUESTIONS

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

Free Response Topics

ArrayList of References / Objects
– get,set,remove,add,size – levels of abstraction

Matrix / 2 D Array
– nested loops, GridWorld (grid)

GridWorld or Make a Class
– location, actor, bug, critter, grid, super, abstract

String / Array Question
– find biggest, find smallest, etc.

© A+ Computer Science - www.apluscompsci.com

Hodge Podge

One question on the A test free response is usually a random question that is hard to predict.

CustomerSort
Robot
Reservation

© A+ Computer Science - www.apluscompsci.com

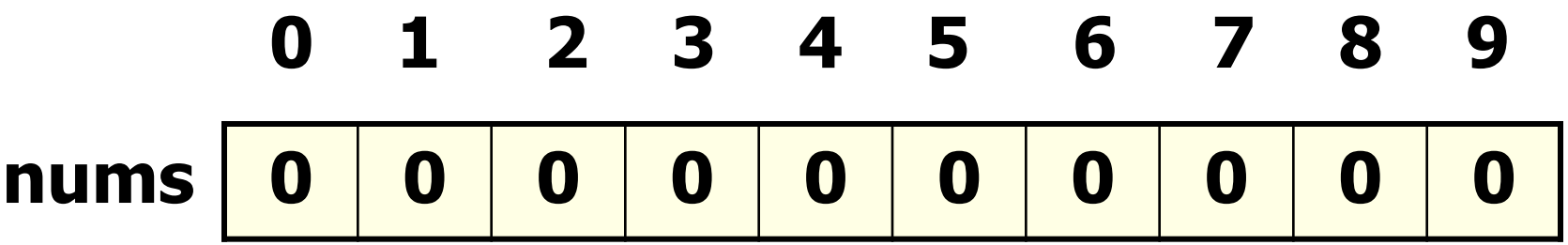
Hodge Podge

This question usually involves an array and many times has sorting and searching components.

© A+ Computer Science - www.apluscompsci.com

Hodge Podge

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

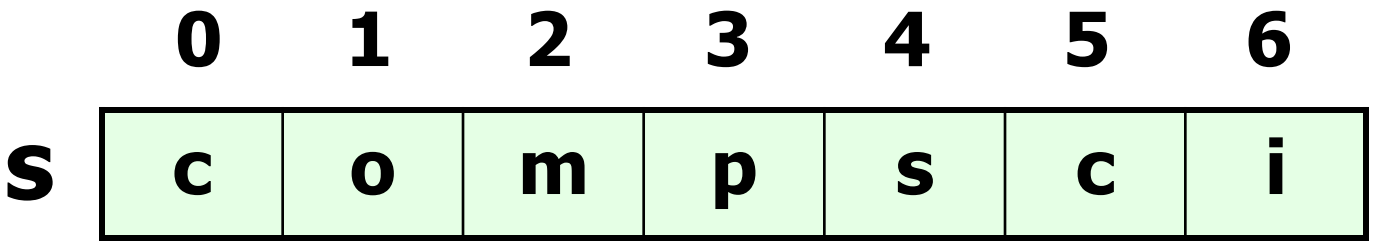


An array is a group of items all of the same type which are accessed through a single identifier.

© A+ Computer Science - www.apluscompsci.com

Hodge Podge

String s = "compsci";



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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

2011 Question 1 - part A

```
public int limitAmplitude(int limit)
{
  int cnt = 0;
  for ( int i = 0; i < samples.length; i++ ) {
    if ( samples[i] < -limit ) {
      samples[i] = -limit;
      cnt++;
    }
    if ( samples[i] > limit ) {
      samples[i] = limit;
      cnt++;
    }
  }
  return cnt;
}
```

2011 Question 1 - part B

```
public void trimSilenceFromBeginning()
{
  int i = 0;
  while ( samples[i] == 0 ) {
    i++;
  }

  int[] ray = new int[ samples.length – i ];
  for (int j = 0; j  < ray.length; j++) {
    ray[j] = samples[j+i];
  }
  samples = ray;
}
```

Abstract / Interfaces

A typical Abstract/Interface question requires that a class be written that extends the abstract class or implements the interface and that all abstract method(s) be implemented.

© A+ Computer Science - www.apluscompsci.com

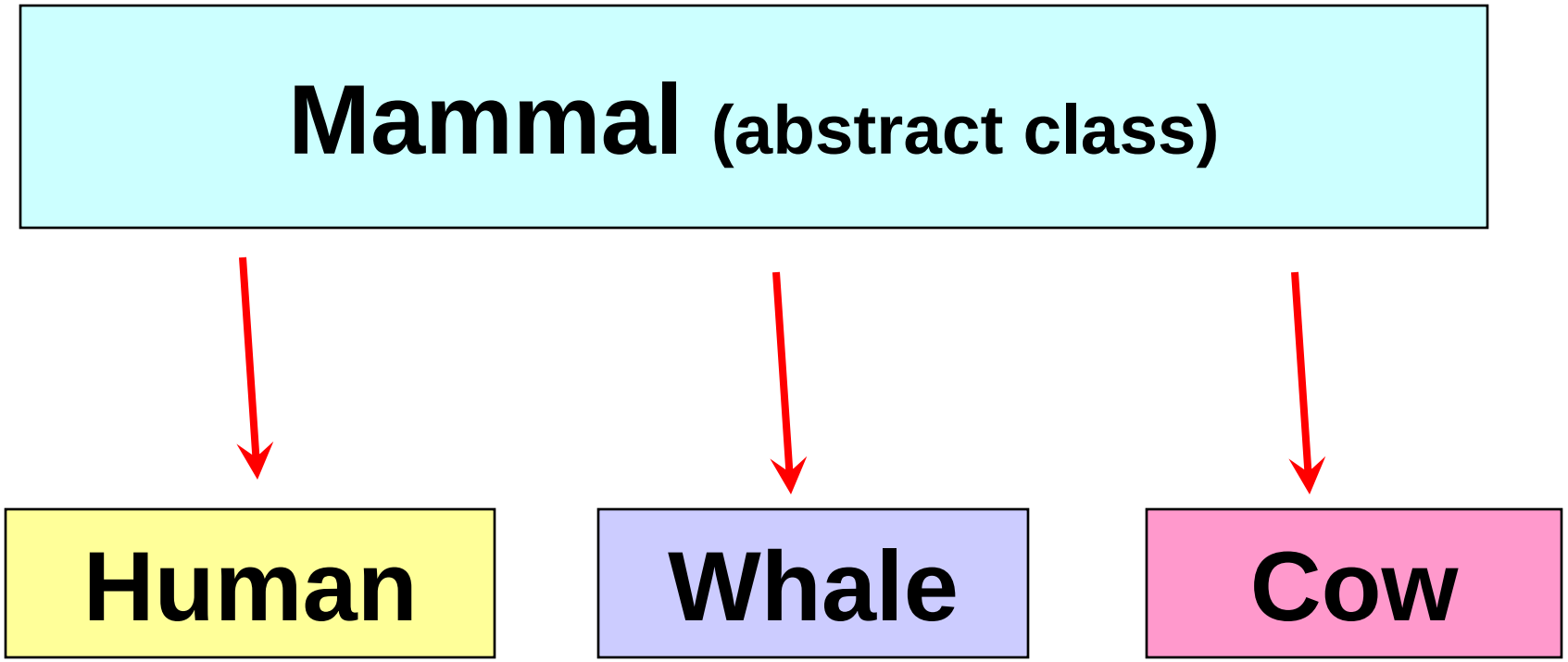
Abstract / Interfaces

Abstract classes are used to define a class that will be used only to build new classes.

No objects will ever be instantiated from an abstract class.

© A+ Computer Science - www.apluscompsci.com

Abstract / Interfaces



© A+ Computer Science - www.apluscompsci.com

Abstract / Interfaces

Any sub class that extends a super abstract class must implement all methods defined as abstract in the super class.

© A+ Computer Science - www.apluscompsci.com

Abstract / Interfaces

```
public abstract class APlus
{
   public APlus(int x)
      //constructor code not shown

   public abstract double goForIt();

   //other fields/methods not shown
}
```

Pet
Item

© A+ Computer Science - www.apluscompsci.com

Abstract / Interfaces

```
public class PassAPTest extends APlus
{
  public PassAPTest(int x)
  {
    super(x);
  }

  public double goForIt()
  {
    double run=0.0;
    //write some code   -   run = x*y/z
    return run;
  }

  //other fields/methods not shown
}
```

```
public abstract class APlus
{
  public APlus(int x)
    //constructor code not shown

  public abstract double goForIt();

  //other fields/methods not shown
}
```

© A+ Computer Science - www.apluscompsci.com

Abstract / Interfaces

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

Methods are public abstract!
Variables are public static final!

© A+ Computer Science - www.apluscompsci.com

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!

© A+ Computer Science - www.apluscompsci.com

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!!!

© A+ Computer Science - www.apluscompsci.com

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.

© A+ Computer Science - www.apluscompsci.com

Abstract / Interfaces

Abstract classes are typically used when you know what you want an Object to do and have a bit of an idea how it will be done.

If the behavior is known and some properties are known, use an abstract class.

© A+ Computer Science - www.apluscompsci.com

2011 Question 2 - part A

```
public class AttractiveCritter extends Critter
{

  public ArrayList<Actor> getActors()
  {
    ArrayList<Actor> stuff;
    stuff = new ArrayList<Actor>();
    for ( Location loc : getGrid().getOccupiedLocations() )
    {
      if ( !loc.equals(getLocation()) )
         stuff.add( getGrid().get(loc) );
    }
    return stuff;
  }
```

You must know ArrayList!

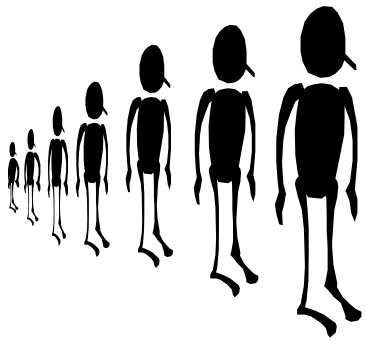
2011 Question 2 - part B

```
public void processActors(ArrayList<Actor> actors)
{
  for (Actor a : actors)
  {
    Location loc = a.getLocation();
    int dir  = loc.getDirectionToward( getLocation() );
    Location x = loc.getAdjacentLocation(dir);
    if (getGrid().get( x ) == null) {
      a.moveTo( x );
    }
  }
}
```

You must know ArrayList!

ArrayList

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



© A+ Computer Science - www.apluscompsci.com

ArrayList

Arraylist is a class that houses an array.

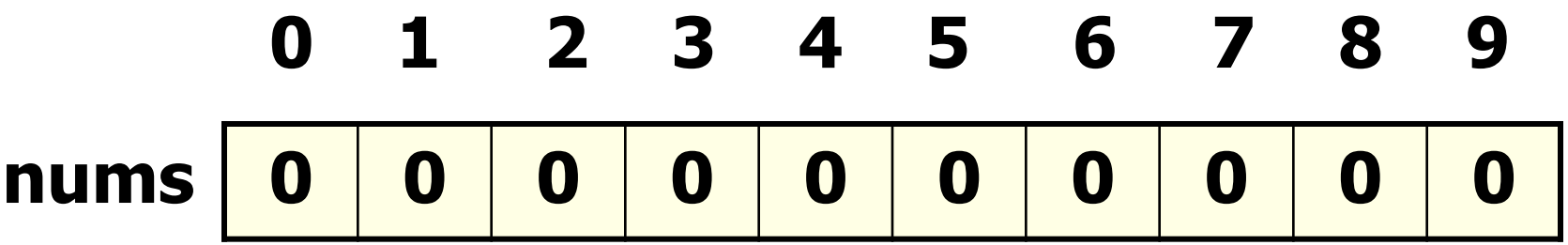
An ArrayList can store any type.

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

© A+ Computer Science - www.apluscompsci.com

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.

© A+ Computer Science - www.apluscompsci.com

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;
```

© A+ Computer Science - www.apluscompsci.com

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.

© A+ Computer Science - www.apluscompsci.com

ArrayList

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

   spot--;

}
```

© A+ Computer Science - www.apluscompsci.com

ArrayList

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

© A+ Computer Science - www.apluscompsci.com

ArrayList

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

© A+ Computer Science - www.apluscompsci.com

2011 Question 3 - part A

```
public int nextTankToFillA(int threshold)
{
  int min = filler.getCurrentIndex();
  for (int i = 0; i < tanks.size(); i++)
  {
    int curr = tanks.get(i).getFuelLevel() ;
    int min = tanks.get(min).getFuelLevel();
    if ( curr <= threshold && curr <  min )
    {
            min = i;
     }
  }
  return min;
}
```

You must know ArrayList!

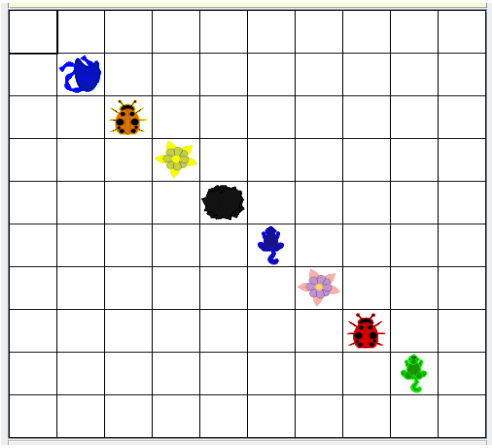
2011 Question 3 - part B

```
public void moveToLocation(int locIndex)
{
  if ( filler.getCurrentIndex() > locIndex) {
    if ( filler.isFacingRight()) {
      filler.changeDirection();
    }
    filler.moveForward(filler.getCurrentIndex() - locIndex);
  }
  if (filler.getCurrentIndex() < locIndex) {
    if (!filler.isFacingRight()) {
      filler.changeDirection();
    }
    filler.moveForward(locIndex - filler.getCurrentIndex());
  }
}
```

You must know ArrayList!

Matrices

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

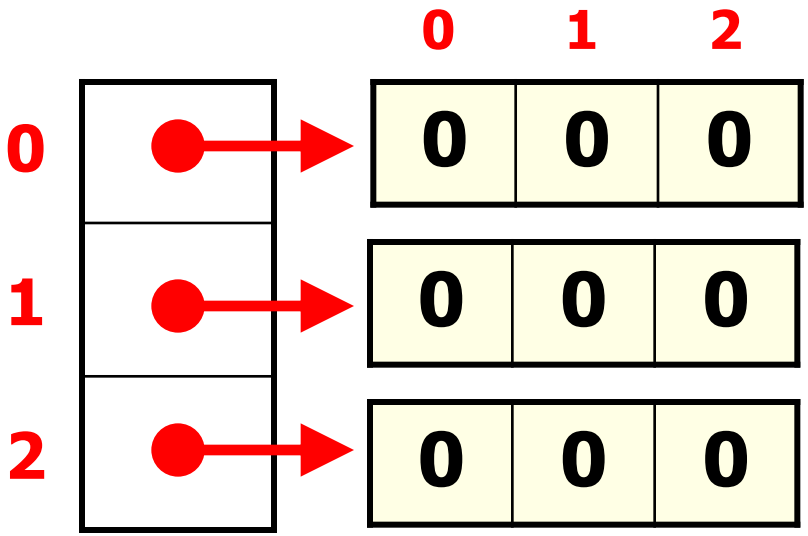


© A+ Computer Science - www.apluscompsci.com

Matrices

A matrix is an array of arrays.

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



© A+ Computer Science - www.apluscompsci.com

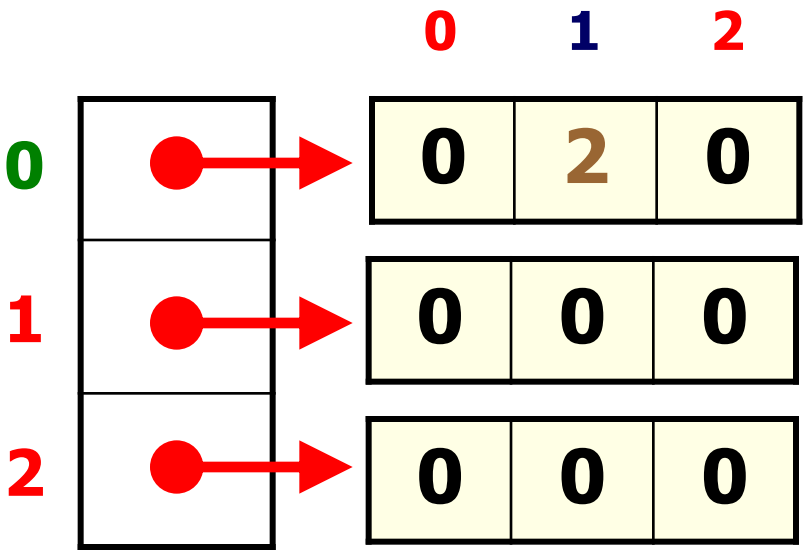
Matrices

A matrix is an array of arrays.

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

Which array?

Which spot?



© A+ Computer Science - www.apluscompsci.com

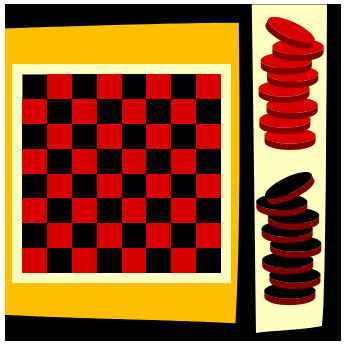
Matrices

	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



© A+ Computer Science - www.apluscompsci.com

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

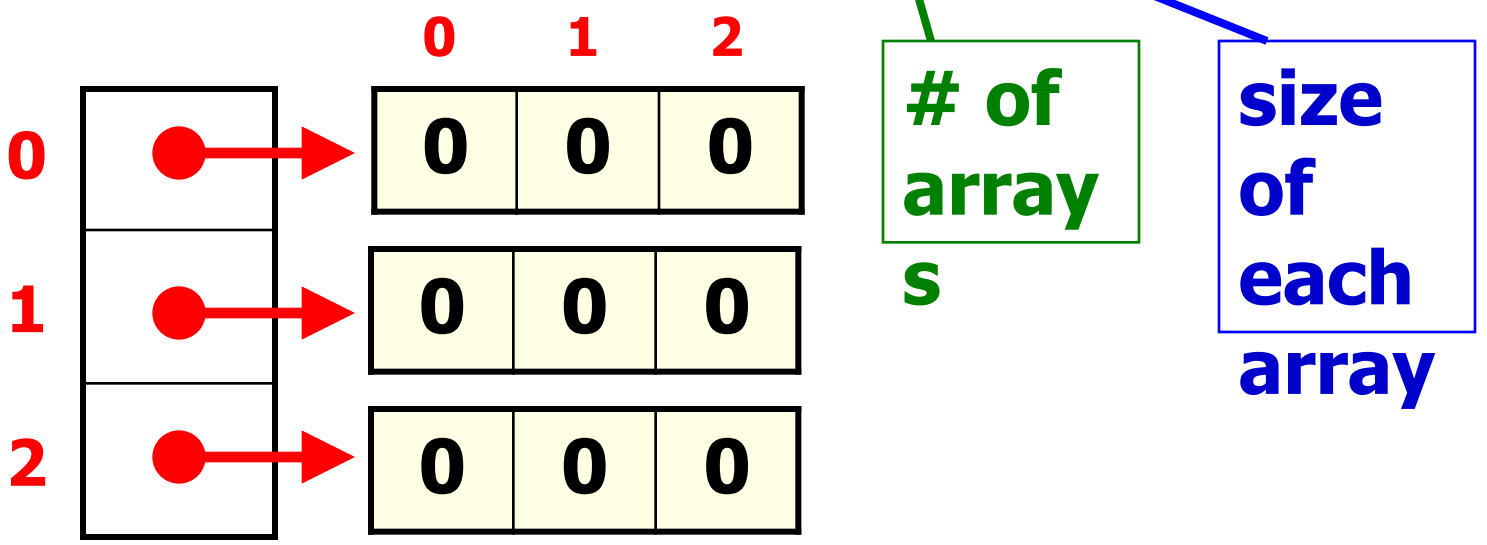
0	0	0
0	1	2
0	2	4

© A+ Computer Science - www.apluscompsci.com

Matrices

A matrix is an array of arrays.

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



© A+ Computer Science - www.apluscompsci.com

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
```

© A+ Computer Science - www.apluscompsci.com

2011 Question 4 - part A

```
private void fillBlock(String str)
{
  int pos = 0;
  for (int r = 0; r < numRows; r++ )
    for (int c = 0; c < numCols; c++ )
   {
      if (pos < str.length()) {
        letterBlock[r][c] = str.substring(pos, pos+1);
        pos++;
      } else{
         letterBlock[r][c] = "A";
      }
   }
}
```

2011 Question 4 - part B

```
public String encryptMessage(String message)
{
  String mess = "";
  int sect = numRows * numCols;
  while (message.length() > 0)
  {
    if (sect > message.length())
       sect = message.length();

    fillBlock(message);
    mess += encryptBlock();
    message = message.substring( sect );
  }
  return mess;
}
```

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

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

© A+ Computer Science - www.apluscompsci.com

Free Response Topics

ArrayList of References / Objects
– get,set,remove,add,size – levels of abstraction

Matrix / 2 D Array
– nested loops, GridWorld (grid)

GridWorld or Make a Class
– location, actor, bug, critter, grid, super, abstract

String / Array Question
– find biggest, find smallest, etc.

© A+ Computer Science - www.apluscompsci.com

A+ Computer Science

AP REVIEW

2011 FR QUESTIONS