# A+ Computer Science

## AP REVIEW

## 2024 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

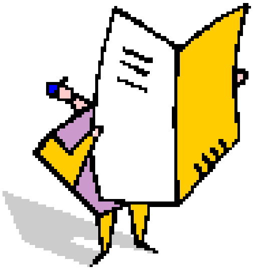
-use the test to take the test

-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

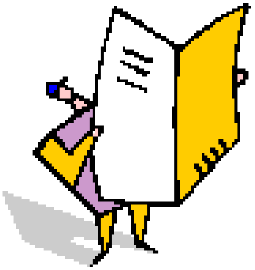
-see if part B calls part A and so on

-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 Topics

## Algorithms / Logic

– ifs, loops, methods

## Make a Class

– create a class

## Array/ArrayList

– get,set,remove,add,size - [],length

## Matrices

– nested loops - array of arrays concepts

# Free Response Question 1

# Algorithms / Logic

# Algorithms / Logic

**Algorithm problems often use array and strings, but like this year, they sometimes just use simple loops and method calls.**

# Algorithms / Logic

```
for(int aplus=1; aplus<7; aplus+=2)
{
  out.println("comp");
  out.println( aplus );
}
```

**OUTPUT**

```
comp
1
comp
3
comp
5
```

# Algorithms / Logic

```
int run=25;
while(run>=10)
{
   out.println(run);
   out.println("loop");
   run=run-5;
}
```

**OUTPUT**

```
25
loop
20
loop
15
loop
10
loop
```

```
public void simulateOneDay(int numBirds)
{
        double r = Math.random()*100;
        if( r < 95.0 )
        {
          int f = (int)(Math.random()*41)+10;
          int eaten = f * numBirds;
          currentFood = currentFood - eaten;
          if( currentFood < 0)
               currentFood = 0;
        }
        else
        {
          currentFood = 0;
        }
}
```

# 2024
# Question 1
# Part A

```
public int simulateManyDays(int numBirds,
                                                     int numDays)
{
          int cnt = 0, x = 0;
          while( x < numDays && currentFood != 0 )
          {
             simulateOneDay(numBirds);
             cnt++;
             x++;
          }
          return cnt;
}
```

# 2024
# Question 1
# Part B.1

```
public int simulateManyDays(int numBirds,
                            int numDays)
{
   int cnt = 0;
   for( int x = 0; x < numDays; x++)
   {
      if( currentFood == 0 )
            break;
      simulateOneDay(numBirds);
      cnt++;

   }
   return cnt;
}
```

**2024**
**Question 1**
**Part B.2**

# Free Response Question 2

# 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.

## Make a Class

## 2024 Question 2

```
public class Scoreboard
{
        private String team1;
        private String team2;
        private int score1;
        private int score2;
        private boolean active1;
        private boolean active2;

        public Scoreboard( String t1, String t2 )
        {
                team1 = t1;
                team2 = t2;
                score1 = score2 = 0;
                active1 = true;
                active2 = false;
        }
```

# Make a Class

```
public String getScore()
{
      return "" + score1 + " - " + score2 +
                 " " + (active1 ? team1 : team2);
}
```

## 2024 Question 2

# Make a Class

```
public void recordPlay( int x )
{
      if( x > 0 )
      {
        if( active1 )
            score1 += x;
        if( active2 )
            score2 += x;
      }
      else
      {
        active1 = !active1;
        active2 = !active2;
      }
}
```

## 2024 Question 2

# Free Response Question 3

# ArrayList

# ArrayList

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

| | | | | | |
|---|---|---|---|---|---|
| 34 | 76 | -8 | 44 | 22 | -998 |

# 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.

| | | | | | |
|---|---|---|---|---|---|
| 34 | 76 | -8 | 44 | 22 | -998 |

# 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 boolean isWordChain()
{
  for( int i = 0; i < wordList.size()-1; i++)
  {
    if( !(wordList.get(i+1).contains(
                            wordList.get(i))))
       return false;

  }
  return true;
}
```

# 2024 Question 3 Part A

```
public ArrayList<String> createList(String target)
{
        ArrayList<String> aplus;
        aplus = new ArrayList<>();
        for( String s : wordList )
        {
          if(s.startsWith(target))
              aplus.add(s.substring(target.length()));
        }
        return aplus;
}
```

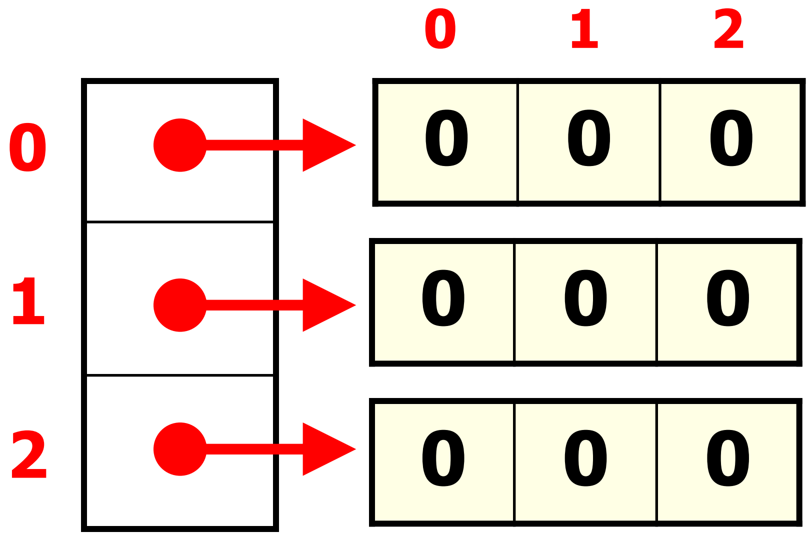
# 2024
# Question 3
# Part B

# Free Response Question 4

# Matrices

# Matrices

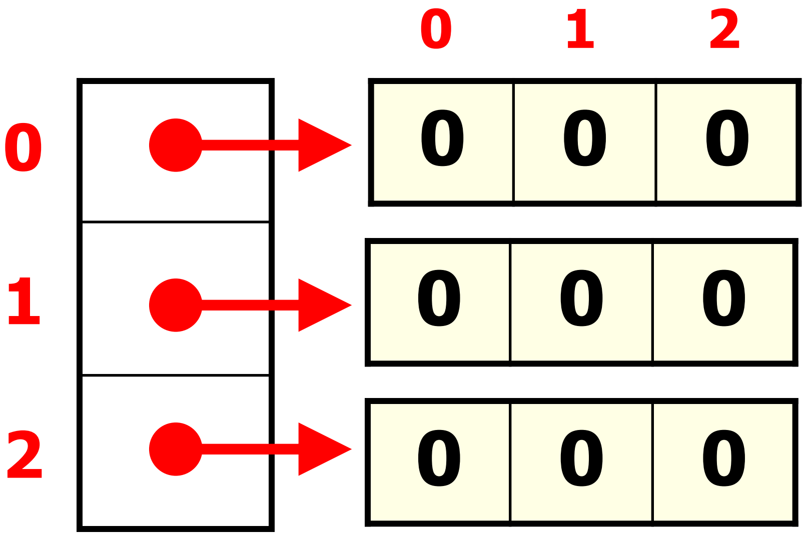
Typically, 1 question on the A test free response will require that students manipulate a 2-dimensional array.



# Matrices

A matrix is an array of arrays.

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



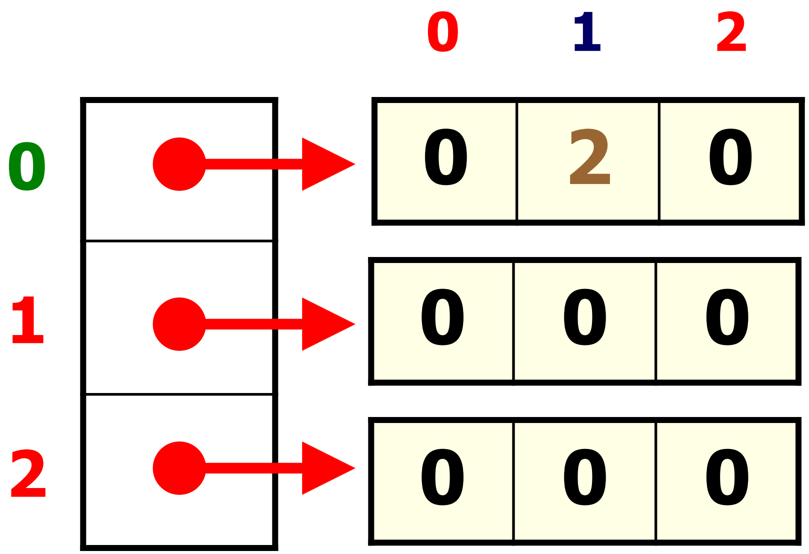
# Matrices

A matrix is an array of arrays.

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

Which array?

Which spot?



| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 0 | 2 | 0 |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 |

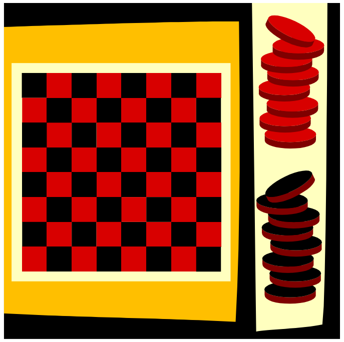
# Matrices

|   | 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 |

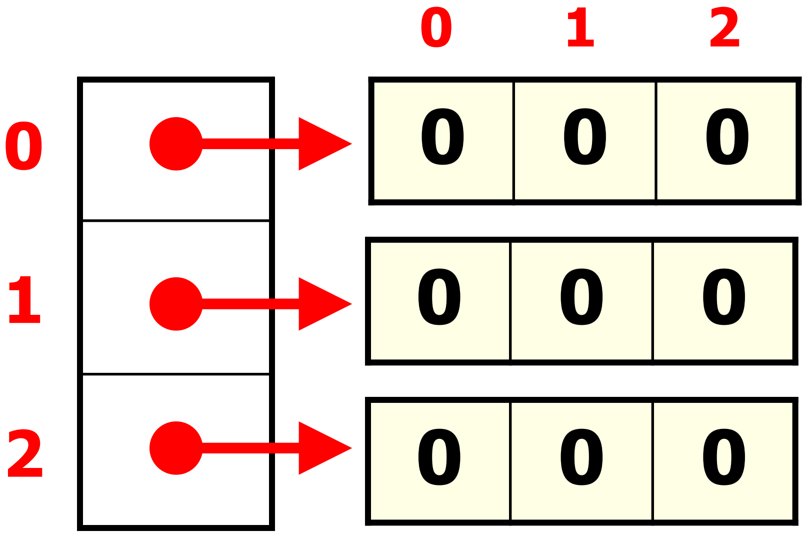
# Matrices

A matrix is an array of arrays.

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

# of arrays

size of each array



# 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 Location getNextLoc(int row, int col){
        Location bel = null, rt = null;
        int bv = 0, rv = 0;
        if( row+1 < grid.length){
          bel = new Location(row+1,col);
          bv = grid[row+1][col];
        }
        if( col+1 < grid[row].length){
                rt = new Location(row,col+1);
                rv = grid[row][col+1];
        }
        if( bel == null )
                return rt;
        if( rt == null )
                return bel;
        if( bv < rv )
                return bel;
        return rt;
}
```

# 2024
# Question 4
# part A

```
public int sumPath(int row, int col)
{
        int sum = 0;
        int rt = grid[row].length-1;
        int bot = grid.length-1;
        sum += grid[row][col];
        while(!(row == bot && col == rt))
        {
                Location nxt = getNextLoc(row,col);
                row = nxt.getRow();
                col = nxt.getCol();
                sum += grid[row][col];
        }
        return sum;
}
```

**2024**
**Question 4**
**part B .1**

```
public int sumPath(int row, int col)
{
        int rt = grid[row].length-1;
        int bot = grid.length-1;

        if(!(row == bot && col == rt))
        {
          Location nxt = getNextLoc(row,col);
          return grid[row][col] +
               sumPath(nxt.getRow(), nxt.getCol());
        }
        return grid[row][col];
}
```

**2024**
**Question 4**
**part B .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

-work through the test more than once

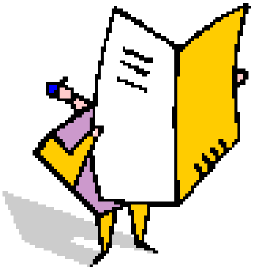
-use the test to take the test

-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

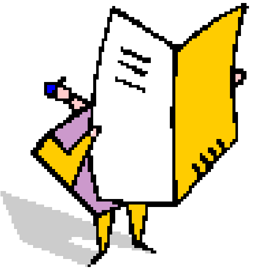
-see if part B calls part A and so on

-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 Topics

## Algorithms / Logic

– ifs, loops, methods

## Make a Class

– create a class

## Array/ArrayList

– get,set,remove,add,size - [],length

## Matrices

– nested loops - array of arrays concepts

# A+ Computer Science

AP REVIEW

2024 AP CS A EXAM