SAMPLE SHORT ANSWER QUESTIONS

WINTER 2019 FINAL EXAM SHORT ANSWER Q7

[2] 7) What is the output of the following code?

```
for ( int i = 200; i > 0; i=i-50 ) {
    println( i );
}
```

Output:

WINTER 2019 FINAL EXAM SHORT ANSWER Q7

[2] 7) What is the output of the following code?

```
for ( int i = 200; i > 0; i=i-50 ) {
    println( i );
}
```

Output:

```
200
150
100
50
```

FALL 2018 FINAL EXAM SHORT ANSWER Q7

[2] 7) Write a loop that will print every Nth number from 0 to 20, inclusive. For example, if N=5, it should print 0, 5, 10, 15, and 20, and if N=6 it should print 0,6,12,18.

FALL 2018 FINAL EXAM SHORT ANSWER Q7

[2] 7) Write a loop that will print every Nth number from 0 to 20, inclusive. For example, if N=5, it should print 0, 5, 10, 15, and 20, and if N=6 it should print 0,6,12,18.

```
for (int i=0; i<=20; i+=N){
   println(i);
}
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q8

[2] 8) Assume the following three functions already exist: `generatePoint()` assigns values to x and y, `offCanvas()` returns true if (x,y) is off the canvas, and `drawCreature()` draws a creature at (x,y). Fill in the blanks below with an appropriate loop that draws creatures until the creature would be off the canvas:

```
int x, y; //state variables
boolean isOnCanvas;

_______                          ___________ {
    generatePoint(); //puts values in x and y
    isOnCanvas = !offCanvas();
    if ( isOnCanvas )
        drawCreature(); //draws creature at x,y

} ______                         _________
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q8

[2] 8) Assume the following three functions already exist: `generatePoint()` assigns values to x and y, `offCanvas()` returns true if (x,y) is off the canvas, and `drawCreature()` draws a creature at (x,y). Fill in the blanks below with an appropriate loop that draws creatures until the creature would be off the canvas:

```
int x, y; //state variables
boolean isOnCanvas;

_____do_____ {
    generatePoint(); //puts values in x and y
    isOnCanvas = !offCanvas();
    if ( isOnCanvas )
        drawCreature(); //draws creature at x,y

} _____while (isOnCanvas);_____
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q10

[2] 10) What is printed by the following code?

```
String s = "";
s += 'c';
s += (4 + 1);
println( s + s.charAt(1) );
```

Output:

WINTER 2019 FINAL EXAM SHORT ANSWER Q10

[2] 10) What is printed by the following code?

```
String s = "";
s += 'c';
s += (4 + 1);
println( s + s.charAt(1) );
```

Output:

```
c55
```

FALL 2018 FINAL EXAM SHORT ANSWER Q11

[2] 11) Complete the function below, so that it is named **maxMagnitude**, accepts two **float** parameters, and returns the one with the larger magnitude (i.e. absolute value).

```
_____                                                    _____
                  if(abs(x) > abs(y))
_____________________                                    _____
                  else
_____________________                                    _____
_____                                                    _____
```

FALL 2018 FINAL EXAM SHORT ANSWER Q11

[2] 11) Complete the function below, so that it is named **maxMagnitude**, accepts two **float** parameters, and returns the one with the larger magnitude (i.e. absolute value).

```
float maxMagnitude(float x, float y) {
    if(abs(x) > abs(y))
        return x;
    else
        return y;
}
```

FALL 2018 FINAL EXAM SHORT ANSWER Q12

[2] 12) What are the two main reasons for defining functions in your program?

_____ _____

_____ _____

FALL 2018 FINAL EXAM SHORT ANSWER Q12

[2] 12) What are the two main reasons for defining functions in your program?

Re-use common code

Readability

FALL 2018 FINAL EXAM SHORT ANSWER Q13

[2] 13) Write ***one*** line of code that creates an array named **`favourites`** that contains three **`String`** values giving the names of your three favourite foods.

FALL 2018 FINAL EXAM SHORT ANSWER Q13

[2] 13) Write ***one*** line of code that creates an array named **`favourites`** that contains three **`String`** values giving the names of your three favourite foods.

```
String[] favourites = {"steak", "lasagna", "apple pie"};
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q13

[2] 13) Given the declaration `int[] arrA = {8, 3, 0, 2, 7, 9, 6};` and the declaration `int[] arrB = new int[10];`, what is the result of each of the following expressions? If an expression will give an error, state the error.

```
arrB[2]                  _____

arrA[arrA.length/3]      _____

arrB[arrB.length]        _____

arrA[arrA[3]]            _____
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q13

[2] 13) Given the declaration `int[] arrA = {8, 3, 0, 2, 7, 9, 6};` and the declaration `int[] arrB = new int[10];`, what is the result of each of the following expressions? If an expression will give an error, state the error.

`arrB[2]`	0
`arrA[arrA.length/3]`	0
`arrB[arrB.length]`	index out of bounds
`arrA[arrA[3]]`	0

WINTER 2019 FINAL EXAM SHORT ANSWER Q14

[2] 14) Given the following code:

```
int[] a;
int[] b = new int[30];
for(int i=1; i<=5; i++){
   b = new int[i];
}
```

How many array variables are created? _____

How many array objects are created? _____

WINTER 2019 FINAL EXAM SHORT ANSWER Q14

[2] 14) Given the following code:

```
int[] a;
int[] b = new int[30];
for(int i=1; i<=5; i++){
    b = new int[i];
}
```

How many array variables are created? __2_____

How many array objects are created? __6_____

FALL 2018 FINAL EXAM SHORT ANSWER Q15

[2] 15) You need to be able to store up to 10 names, using a partially-full array. Give the ***declaration statements*** that would define and initialize all of the variables and constants that you would need to do this. ***Do not write any other code.***

FALL 2018 FINAL EXAM SHORT ANSWER Q15

[2] 15) You need to be able to store up to 10 names, using a partially-full array. Give the ***declaration statements*** that would define and initialize all of the variables and constants that you would need to do this. ***Do not write any other code.***

```
final int MAX_SIZE = 10;
String[] names = new String[MAX_SIZE];
int numNames = 0;
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q15

[2] 15) Suppose we declared a partially-filled array (PFA) of ints:

```
final int MAX_PFA_SIZE = 100;
int[] myPFA = new int[ MAX_PFA_SIZE ];
int currentSize = 0; //current number of ints stored in myPFA
```

Fill in the blanks so that the following code prints on the console all numbers currently stored in myPFA that are bigger than 15:

```
for ( int i = 0; ____          _____; ____          ___) {

    if (_________              ____________________)

        println(_____                  __________);
} // end for
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q15

[2] 15) Suppose we declared a partially-filled array (PFA) of ints:

```
final int MAX_PFA_SIZE = 100;
int[] myPFA = new int[ MAX_PFA_SIZE ];
int currentSize = 0; //current number of ints stored in myPFA
```

Fill in the blanks so that the following code prints on the console all numbers currently stored in myPFA that are bigger than 15:

```
for ( int i = 0; _____i < currentSize_____; _________i++_____) {

    if (_________myPFA[i]>15_____________________________)

        println(_____myPFA[i]_____________________________);
} // end for
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q16

[2] 16) Complete the following function to delete an element at a given position from a partially-filled array.

```
int delete( int[] data, int size, int position){
   //delete the item in the given position, if the position is OK
   if(_________                                    _________________){
      for(int i=__          ____;__          ______; i++){
         data[i] = data[i+1];
      }
      size--;
   }
   return size;
}
```

WINTER 2019 FINAL EXAM SHORT ANSWER Q16

[2] 16) Complete the following function to delete an element at a given position from a partially-filled array.

```
int delete( int[] data, int size, int position){
   //delete the item in the given position, if the position is OK
   if(_________position>=0 && position<size_____________){
      for(int i=___position_____;___i<size-1_________; i++){
         data[i] = data[i+1];
      }
      size--;
   }
   return size;
}
```

SAMPLE PROGRAMMING QUESTIONS

WINTER 2019 FINAL EXAM PROGRAMMING Q21

[5] 21) Complete the following non-active program so that the output is a modified version of the `String s`, where all spaces have been replaced with the last vowel before the space. If a space is encountered before any vowels, replace the space with `'a'`. For example, if `s = "Computer science is fun"`, the output should be `"Computerescienceeisifun"`. If `s = "h ve go dsummer"`, the output should be `"haveegoodsummer"`.

```
String s = "There are many vowels in most sentences."
String output = "";

//  FILL IN HERE



println(output);

//  END OF PROGRAM
```

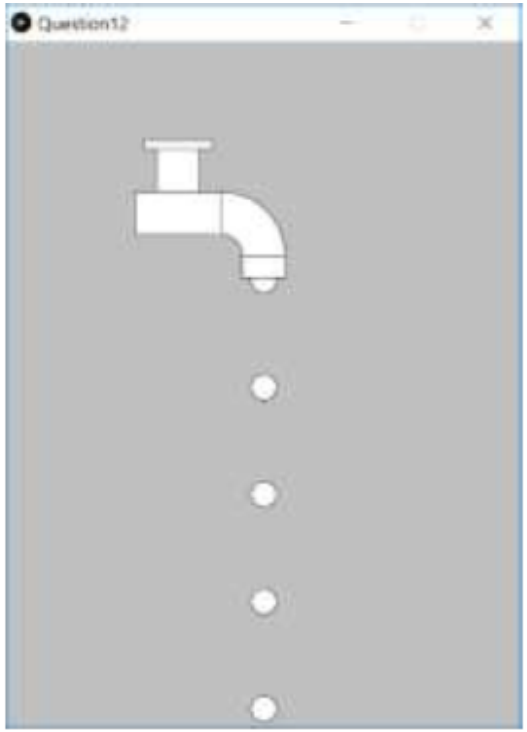
WINTER 2019 FINAL EXAM PROGRAMMING Q21

```
String s = "There are many vowels in most sentences."
String output = "";
char lastVowel = 'a';

for (int i=0; i<s.length(); i++){
  char curr = s.charAt(i);
  if (curr == ' ')
    output += lastVowel;
  else{
    if(curr=='a' || curr=='e' || curr=='i' || curr=='o' || curr=='u')
      lastVowel = curr;
    output += curr;
  }
}
println(output);

//  END OF PROGRAM
```

FALL 2018 FINAL EXAM PROGRAMMING Q20

[5] 20) Complete the program below, which animates drops of water falling from a leaky tap. Assume that the code to draw the tap has already been written. You do not have to write it. You need to draw only the water drops.

- The drops should be white, and they should be in the centre of the canvas horizontally.
- Each drop should be **DROP_SPACING** pixels lower than the one above it.
- The centre of the highest drop should be at **DROP_START** plus a small amount stored in **dropFallen**, which controls how far the top drop has fallen.
- Slowly change the value of **dropFallen** to cause the drops to fall. When **dropFallen** reaches **DROP_SPACING** then you should start a new drop at **DROP_START**.
- The top drop should be hidden inside the faucet when it begins to fall, as shown.



Also see next page

FALL 2018 FINAL EXAM PROGRAMMING Q20

```
final int DROP_START = 250; //Y coordinate where the drop centres start
final int DROP_SPACING = 125; //Space between drops in the Y direction
final int DROP_DIAMETER = 30; //Diameter of all drops
final int DROP_SPEED = 3; //Speed that all drops fall (pixels per frame)

int dropFallen = 0; //How far the top drop has fallen from DROP_START.

void setup(){
  size(600,800);
}
void drawFaucet(){
  //Assume code to draw the faucet is here. You DO NOT need to write anything.
}
void draw(){
  background(192);




}

void drawDrops(){
  //This function should draw all of the drops.
```

FALL 2018 FINAL EXAM PROGRAMMING Q20

```
final int DROP_START = 250; //Y coordinate where the drop centres start
final int DROP_SPACING = 125; //Space between drops in the Y direction
final int DROP_DIAMETER = 30; //Diameter of all drops
final int DROP_SPEED = 3; //Speed that all drops fall (pixels per frame)

int dropFallen = 0; //How far the top drop has fallen from DROP_START.

void setup(){
  size(600,800);
}
void drawFaucet(){
  //Assume code to draw the faucet is here. You DO NOT need to write anything.
}
void draw(){
  background(192);

  drawDrops();
  drawFaucet();

}

void drawDrops(){
  //This function should draw all of the drops.
  fill(255); //optional
  int nextDrop = DROP_START +dropFallen;
  while(nextDrop<height+DROP_DIAMETER/2){
    ellipse(width/2,nextDrop,DROP_DIAMETER,DROP_DIAMETER);
    nextDrop += DROP_SPACING;
  }
  dropFallen = (dropFallen +DROP_SPEED)%DROP_SPACING;
}
```

WINTER 2019 FINAL EXAM PROGRAMMING Q23

[5] 23) Write a function named `filter`, which will accept an array of `int` values, and an `int` named `min`. It should create and return another array of `int` values, where the new array contains all values from the original array that are greater than or equal to `min`. For example, for the array

```
int[] test ={3,9,-2,6,1,8};
```

the function call

```
int[] result = filter(test,4);
```

should return the array

```
{9,6,8}.
```

The array returned should be exactly the correct size to hold the required number of values. The original array may be any size, 0 or greater.

WINTER 2019 FINAL EXAM PROGRAMMING Q23

```
int[] filter(int[] x, int min){
  //count the number of entries greater than or equal to min
  int count=0;
  for (int i=0; i<x.length; i++)
    if (x[i] >= min)
      count++;

  //create new array
  int[] y = new int[count];

  //fill new array
  int n = 0;
  for (int i=0; i<x.length; i++) //loop over original array
    if (x[i] >= min) {
      y[n] = x[i]; //copy from orig to next empty bin in new array
      n++;
    }
  return y;
}
```

FALL 2018 FINAL EXAM PROGRAMMING Q23

[5] 23) Complete the function **validatedData(float[] data)** which will accept a full array of **float** values. It should create and return another array of **float** values containing only the "valid" data values from the array. The **validate** function (shown below) must be used to determine if a value is valid or not. The array returned should be exactly the correct size to hold the required number of values.

```
boolean validate(float x){ return 0 <= x && x <= 100; }

float[] validatedData(float[] data){
```

FALL 2018 FINAL EXAM PROGRAMMING Q23

[5] 23) Complete the function **validatedData(float[] data)** which will accept a full array of **float** values. It should create and return another array of **float** values containing only the "valid" data values from the array. The **validate** function (shown below) must be used to determine if a value is valid or not. The array returned should be exactly the correct size to hold the required number of values.

```
boolean validate(float x){ return 0 <= x && x <= 100; }

float[] validatedData(float[] data){
  int counter = 0;
  for(int i = 0; i < data.length; i++)
    if(validate(data[i]))
      counter++;
  float [] result = new float[counter];
  counter = 0;
  for(int i = 0; i < data.length; i++)
    if(validate(data[i]))
      result[counter++] = data[i];
  return result;
}
```

WINTER 2019 FINAL EXAM PROGRAMMING Q24

[5] 24) Complete the function `myBestSearch` below, which should search the sorted array `myArray` for `key`. If `key` is found, the function should return the index (position) where it occurs in `myArray`. If the `key` is not found in the array, the function should return `-1`.
Note: The maximum mark is 5 if you do a binary search, and 3 if you do a linear search.

```
int myBestSearch (int[] myArray, int key){
```

WINTER 2019 FINAL EXAM PROGRAMMING Q24

```
int myBestSearch (int[] myArray, int key){
    int lo=0;
    int hi=myArray.length-1;
    int mid;
    int foundPosn = -1;

    while(lo<=hi){
       mid = lo + (hi-lo)/2;
       if(myArray[mid]==key){
          foundPosn = mid;
          lo = hi + 1;
       }
       else if(myArray[mid]<key)
          lo = mid + 1;
       else
          hi = mid - 1;
    }//while

    return foundPosn;
} //myBestSearch
```