CSCI1530 Computer Principles and Java Programming

Tutorial 12

Final Revision

Question 1:

- Complete the following method for determining if the parameter contains the following substring "123". If yes, return true; otherwise, return false. If the parameter is null or is an empty String, also return true.
- public Boolean contains123 (String s){
  }

Answer 1:

Analysis:

1.How to get substring --------- indexOf()

2.How to determine a string is null string or empty string

-----null string(null)/empty string("")

```
public boolean contains123(String s){
        if(s==null || s.length()==0)
            return true;
        else if (s.indexOf("123")!=-1)
            return true;
        else
            return false;
    }
```

Question 2:

- What are the differences in syntax form and in function between a constructor and an ordinary method?
- Answer:

Syntax:
(1) the name of constructor must be same as name of the class but there is no such requirement for method in Java. Methods can have any arbitrary name in java.
(2) constructor has no return type but methods do bear a type or void.

Function:
(1) constructor is for initializing a new object. Method is for performing some task on/for an object.

Question 2:

- What is the compilation error?

```
class ExamTime {
	int v = 30;
	public static void main( String[] args ) {
		System.out.println("value is " + v);
	}
}
```

- Answer:

Non-static variable v cannot be referenced from a static context.

Correction:

```
ExamTime obj = new ExamTime();
System.out.println(obj.v);
```

Question 2:

- Let SpaceCar be a class that has been properly defined

```
SpaceCar car1 car2;
car 1 = new SpaceCar();
car 2 = car1;
```

```
SpaceCar car1 car2;
car 1 = new SpaceCar();
car 1 = new SpaceCar();
car 2 = new SpaceCar();
```

- Ask: How many SpaceCar objects are created in the above?
- Answer: 1 and 3 respectively

Hint: Number of new SpaceCar() == Number of objects

Question 2:

- What will be printed by each of the following code:

```
public class exam{
        public static void main (String args[]) {
        String s1 = "ABC";
        String s2 = new String("ABC");
       if (s1 ==s2)
               System.out.println(1);
       else
               System.out.println(2);
```

- Analysis:
  `String s1 = "ABC";` ----------String literal
  String s2 = new String("ABC");----String object
- Answer: 2

Question 2:

- Similar to the last item:

```
public class exam{
        public static void main (String args[]) {
        String s1 = "ABC";
        String s2 = "ABC";
       if (s1 ==s2)
               System.out.println(1);
       else
               System.out.println(2);
```

- Analysis:

  Both s1,s2 refer to the same String literal.

- Answer: 1

Question 2:

```
public class exam{
        public static void main (String args[]) {
       int i =  3/2;
       switch(i) {
               case 0: System.out.println("aaa"); break;
               case 1: System.out.println("bbb");
               case 2: System.out.println("ccc"); break;
        }
}
```

- Answer : bbb

  ccc

Question 2:

```
public class exam{
        public static void main (String args[]) {
                int i = 3%2;
                do{
                    i--;
                } while (i > 2);
                System.out.println(i);
        }
}
```

- Answer: `0`
- Hint: do{…} while(condition) : at least one iteration

Question 2:

```
public class exam{
        public static void main{String args[]} {
                int i;
                double d = 3.7;
                i = ((int)d ) * ((int)Math.round(d));
                System.out.println(i);
        }
}
```

- Answer: 12

(int) d = 3

(int)Math.round(d) = 4

Question 2:

```
int count = 1;
while ( count != 10 ) {
        count = count + 2;
}
System.out.println(count);
```

- Answer: i never reaches 10

  Endless loop….

Question 3:

```
public class Stars {
        Public static void main (String[] args) {
                int max = 5;
                for (lint row =1; row <= max; row++) {
                        for (int star = 1; star <= row; star++)
                                System.out.print("*");
                        System.out.println();
                }
        }
}
```

- Answer:

```
*
**
***
****
*****
```

Question 4:

- Rewrite the following nested if statement into an equivalent switch statement.

```
if (number == 5)
    myChar = 'A';
else
    if (number == 6)
        myChar = 'B' ;
    else
       myChar = 'C' ;
```

```
switch(number){
        case 5:
                myChar = 'A';
                break;
        case 6:
                myChar = 'B';
                break;
        default:
                myChar = 'C';
                break;

 }
```

Question 5:

- print the prime numbers in array pnumbers in reverse order. You should use the array attribute "length".

```
public class Foo {
        public static void main (String[] args) {
                int[] pnumbers = {2, 3, 5, 7, 11, 13, 17, 19};
        }
}
```

- Hint:

Array attribute: pnumbers.length;

Array index: [0, length-1]

Array element can be accessed by variable index: for loop

Answer 5:

```
public class Foo {
        public static void main (String[] args) {
                int[] pnumbers = {2, 3, 5, 7, 11, 13, 17, 19};
                int size = pnumber.length;
                for (int i=size-1; i>=0; i--) {
                        System.out.println(pnumber[i]);
                }
        }
}
```

Question 6:

- Write a method that computes the following for integers a and n.

$$a^n + a^{n-1} + \cdots + a^2 + a$$

- The method accepts integers a and n as parameters and returns the result as type long, assume the result will not exceed the maximum value of a long type.
- Analysis:

```
public static long method_name(int a,int n){
   ……
}
```

Answer 6:

```
public static long expsum(int a, int n) {
        long result = 0;
        int power_a = 1;
        for (int i=1; i<=n; i++) {
            power_a = power_a * a;
            result = result + power_a;
        }
        return result;
}

// alternatively: use Geometric Progression…
```

Question 7:

- print a giant pattern, integer N which is assumed to have been initialized. The method prints nothing and returns immediately if N is negative.

Four sample runs on different values of N:

M	\/	\ / \/	\ / \ / \ / \/
N = 0	N = 1	N = 2	N = 5

Question 7:

```
public static void printPattern(int N){
       if(N<0) return;
       if(N==0) System.out.println('M'); return;
        for(int i=0;i<N;i++){          //N line
            for(int j=0;j<2*N+2;j++){ //2*N+1 symbols each line
                if(j==0||j==2*N+1)
                     System.out.print('|');
                if(j==i+1)
                    System.out.print('\\');
                if(j==(2*N+1-i-1))
                    System.out.print('/');
                else
                    System.out.print(' ');
            }
            System.out.println(); }
    }
```

Question 8:

- Given 2 USB storage devices, each of capacity m, and 6 computer files of sizes `s1,s2,s3,s4,s5,s6`, where

  `s1 + s2 + s3 +s4 + s5 +s6 < 2m.`

- Write a method `public Boolean canPack (int [] s, int m)` to determine whether the 6 files can be stored into the 2 devices, where each file must be stored completely in one of the 2 devices. **`canPack`** accepts parameters m and an array s of integers for sl, s2, ..., s6, and returns true if the files can be stored, and returns false otherwise. Also write any method that `canPack` calls.

Analysis 8:

- 6 files, each file can either be in the first USB device or second USB device. For each file

  `0`: in the first usb device

  `1`: in the second usb device

Brute force searching :

from `000000` to `111111`

```
For example:
001001 : s1,s2,s4,s5 in the first usb device
  ↑  ↑   if s1+s2+s4+s5<m && s3+s6 <m
         return true
```

```
public static Boolean canPack (int [] s, int m){
        for(long i=0;i<64;i++){
            String str = Long.toBinaryString(i);
            int length;
            length = str.length();
            for(int j=6-length;j>0;j--){ //fill zeros
                str = "0"+str;
            }
            int usb1 = 0;
            int usb2 = 0;
            for(int j=0;j<6;j++){
                if(str.charAt(j)=='0')
                    usb1 = usb1+s[j];
                else
                    usb2 = usb2+s[j];
            }
            if(usb1<m && usb2<m)
                return true;
        }
        return false;
    }
```