TAT NO. : 2 Date :

Output Verification : 10 Marks each Total Marks: 120 Marks

Time :1 hr 30 min.

1. Write a java program to find the solution for following task.We have a loud talking parrot. The "hour" parameter is the current hour time in the range 0 to 23. We are in trouble if the parrot is talking and the hour is before 7 or after 20. Return true if we are in trouble.

2. Write a java program to find the solution for following task .The parameter weekday is true if it is a weekday, and the parameter vacation is true if we are on vacation. We sleep in if it is not a weekday or we're on vacation. Return true if we sleep in.

3. Given an int n, return true if it is within 10 of 100 or 200. Note: Math.abs(num) computes the absolute value of a number.

4. Given two non-negative int values, return true if they have the same last digit, such as with 27 and 57. Note that the % "mod" operator computes remainders, so 17 % 10 is 7.

5. Write a static method majority() that takes three boolean arguments and returns true if at least two of the arguments have the value true, and false otherwise. Do not use an if statement.

6. Write a static method max() that takes three int values as arguments and returns the value of the largest one. Add an overloaded function that does the same thing with three double values.

7. Add an abstract *int avgBreedWeight()* method to the Dog class. Remember that this means that the word *abstract* appears in the method header after *public*, and that the method does not have a body (just a semicolon after the parameter list). It makes sense for this to be abstract, since Dog has no idea what breed it is. Now any subclass of Dog must have an avgBreedWeight method; since both Yorkshire and Laborador do, you should be all set.Save these changes and recompile DogTest.java. You should get an error in Dog.java (unless you made more changes than described above). Figure out what's wrong and fix this error, then recompile DogTest.java. You should get another error, this time in DogTest.java. Read the error message carefully; it tells you exactly what the problem is. Fix this by changing DogTest (which will mean taking some things out).

```
// ************************************************************
// Dog.java
//
// A class that holds a dog's name and can make it speak.
//
// ************************************************************
public class Dog
```

```
{
protected String name;
// ----------------------------------------------------------
// Constructor -- store name
// ----------------------------------------------------------
public Dog(String name)
{
this.name = name;
}
// ----------------------------------------------------------
// Returns the dog's name
// ----------------------------------------------------------
public String getName()
{
return name;
}
// ----------------------------------------------------------
// Returns a string with the dog's comments
// ----------------------------------------------------------
public String speak()
{
return "Woof";
}

}
*********************************************************************
// Labrador.java
//
// A class derived from Dog that holds information about
// a labrador retriever. Overrides Dog speak method and includes
// information about avg weight for this breed.
//
// ******************************************************************
public class Labrador extends Dog
{
private String color; //black, yellow, or chocolate?
private int breedWeight = 75;
public Labrador(String name, String color)
{
this.color = color;
}
// ----------------------------------------------------------
// Big bark -- overrides speak method in Dog
// ----------------------------------------------------------
public String speak()
{
return "WOOF";
```

```
}
// -------------------------------------------------------
// Returns weight
// -------------------------------------------------------
public static int avgBreedWeight()
{
return breedWeight;
}
}
//*************************************************************
// Yorkshire.java
//
// A class derived from Dog that holds information about
// a Yorkshire terrier. Overrides Dog speak method.
//
// ************************************************************
public class Yorkshire extends Dog
{
public Yorkshire(String name)
{
super(name);
}
// -------------------------------------------------------
// Small bark -- overrides speak method in Dog
// -------------------------------------------------------
public String speak()
{
return "woof";
}
}
// ************************************************************
// DogTest.java
//
// A simple test class that creates a Dog and makes it speak.
//
// ************************************************************
public class DogTest
{
public static void main(String[] args)
{
Dog dog = new Dog("Spike");
System.out.println(dog.getName() + " says " + dog.speak());
}
}
```

8. Given 2 positive int values, return the larger value that is in the range 10 to 20 inclusive, or return 0 if neither is in that range.
9. We'll say that a number is "teen" if it is in the range 13 to 19 inclusive.Given 3 int values, return true if 1 or more of them are teen.
10. Write a java program Given two int values, return their sum. Unless the two values are the same, then return double their sum.

11. Implement a Java-main-method that prints out the multiplication table for all numbers from 1 to 10. Use the tabulator character '\t' to align the values. The output of your method should be as follows:

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

12. Implement a Java-method with three local int variables a, b, and c that sorts these three values in ascending order by comparing and exchanging their values. At the end of the program a <= b <= c must hold. Don’t use any functions.You can solve this exercises with three if in minimum. Test your method with all six permutation of three different numbers.