Multiple choice questions: [5 points each]

1. A cast is required in which of the following situations?
   a) using charAt to take an element of a String and store it in a char
   b) storing an int in a float
   c) storing a float in a double
   **d) storing a float in an int**
   e) all of the above require casts

2. What will be the result of the following assignment statement? Assume b = 5 and c = 10.
   \[
   \text{int } a = b \times (-c + 2) / 2 / 2;
   \]
   a) 40
   b) −40
   c) 10
   **d) −10**
   e) −3

3. If you want to store into the String name the value “Ghengis Khan”, you would do which statement?
   a) String name = "Ghengis Khan";
   b) String name = new String("Ghengis Khan");
   c) String name = "Ghengis" + " " + "Khan";
   d) String name = new String("Ghengis" + "   " + "Khan");
   **e) Any of the above would work**

4. In Java, “instantiation” means
   a) Noticing the first time something is used
   **b) Creating a new object**
   c) Creating a new alias to an existing object
   d) Launching a method
   e) None of the above
5. Given a String s, which is assumed to have at least one character in it, which of the following conditions would determine if the first character of the String is the same as the last character?
   a) (s.charAt(0) == s.charAt(s.length( )))
   b) (s.charAt(1) == s.charAt(s.length( )))
   c) (s.charAt(0) == s.charAt(s.length( ) - 1))
   d) (s.charAt(0) == s.charAt(s.length( ) + 1))
   e) (s.charAt(0) == s.charAt(last))

6. Given the following code, where x = 0, what is the resulting value of x after the for-loop terminates?
   
   ```java
   for ( int i = 0;  i < 5;  i++ )
   {x += i;}
   ```
   
   a) 0  
b) 4  
c) 5  
**d) 10**  
e) 15

7. An object that refers to part of itself within its own methods can use which of the following reserved words to denote this relationship?
   a) inner 
b) i  
c) private  
**d) this**  
e) static

8. Having multiple class methods of the same name where each method has a different number of or type of parameters is known as
   a) encapsulation  
b) information hiding  
c) tokenizing  
d) importing  
**e) method overloading**
9. Consider the following recursive method:

```java
public int question9(int x, int y)
{
    if (x == y) return 0;
    else return question9(x-1, y) + 1;
}
```

If the method is called as `question9(8, 3)`, what is returned?

- a) 11
- b) 8
- c) 5
- d) 3
- e) 24

10. Each row contains the same value written in different bases. Fill in the missing entries in rows a-c. [14 points]

<table>
<thead>
<tr>
<th>decimal</th>
<th>binary</th>
<th>octal</th>
<th>hex</th>
</tr>
</thead>
<tbody>
<tr>
<td>85</td>
<td>01010101</td>
<td>125</td>
<td>55</td>
</tr>
<tr>
<td>37</td>
<td>00100101</td>
<td>45</td>
<td>25</td>
</tr>
<tr>
<td>129</td>
<td>10000001</td>
<td>201</td>
<td>81</td>
</tr>
</tbody>
</table>
11. Consider the following definition of a palindrome:

String s is a palindrome if the length of s is less than 2.
If the length of s is at least 2, then s is a palindrome if and only if
i) the first and last characters of s are the same, and
ii) the substring between the first and last characters of s is a palindrome.

Write a recursive method based on this definition which determines whether string s is a palindrome. Recall the String methods substring(i,j) - return the substring from position i up to but not including position j - and charAt(i) - retrieve the character at position i. [15 points]

Here are three possible answers

```java
public static boolean palindrome( String s )
{
    if (s.length()<2)
        return true;
    return ((s.charAt(0)==s.charAt(s.length()-1)) &&
            palindrome(s.substring(1,s.length()-1)));
}

public static boolean pal2( String s )
{
    if (s.length()<2)
        return true;
    else
        if ((s.charAt(0)==s.charAt(s.length()-1)) &&
            pal2(s.substring(1,s.length()-1)))
            return true;
        else
            return false;
}
```
public static boolean pal3( String s )
{
    if (s.length()<2)
        return true;

    if ((s.charAt(0)==s.charAt(s.length()-1)))
        return pal3(s.substring(1,s.length()-1));
    else
        return false;
}

12. A **diphthong** is defined (here) as two consecutive vowels (such as "ae" or "oi") in a string. Your job is to write a method which will remove all diphthongs in a given string. You will be provided with a method isVowel, which determines whether a given character is a vowel. Note that if you come across three consecutive vowels, as in the string "xyouiz", you will remove the first diphthong "ou", resulting in "xyiz". However, if there are four vowels in a row, as in "xaeoiyz", you remove both diphthongs "ae" and "oi", leaving "xyz". [16 points]

Here are two solutions:

public static boolean isVowel(char c)
{
    c = Character.toLowerCase(c);

    if (c=='a' || c=='e' || c=='i' || c=='o' || c=='u')
        return true;
    else
        return false;
}
public static String diphthongRemove( String s )
{
    String result = "";
    int i=0;
    while (i<s.length()){
        if ( !isVowel(s.charAt(i)) ||
            (i >= s.length()-1 ) ||
            ( isVowel(s.charAt(i)) && !isVowel(s.charAt(i+1)))
        ){
            result += s.charAt(i);
            i++;
        }
        else
            i+=2;
    }
    return result;
}

public static String diphthongRemove2( String s )
{
    String result = "";
    //We add a space to the end of s to avoid out-of-bounds error
    //It does not get added to result
    s += " ";
    int i=0;
    while ( i< s.length()-1 ) {
        if ( isVowel(s.charAt(i)) && isVowel(s.charAt(i+1)) )
            i += 2;
        else
            {
                result += s.charAt(i);
                i += 1;
            }
    }
    return result;
}
13. Consider the two classes below. SetNum, as seen in class, stores a single (private) integer. IssueConfuser creates two SetNum objects, one int, and one String. It then calls one of its methods which changes (or not) these values in various ways.

What values get printed by the main routine of IssueConfuser? [10 points]

1 6 3 four

--IssueConfuser

public class IssueConfuser
{
    public void confuseIssue(SetNum a, SetNum b, int c, String d)
    {
        a = new SetNum(5);
        b.changeTo(6);
        c = 7;
        d = d.replace('f', 'y');
    }

    public static void main (String[] args)
    {
        IssueConfuser the_confuser = new IssueConfuser();

        SetNum f1 = new SetNum(1);
        SetNum f2 = new SetNum(2);
        int f3 = 3;
        String f4 = "four";

        the_confuser.confuseIssue(f1, f2, f3, f4);

        System.out.println(f1+" "+f2+" "+f3+" "+f4);
    }
}
--SetNum

public class SetNum
{
    private int storedNum;

    public void changeTo(int newVal)
    { this.storedNum = newVal; }

    public int returnNum()
    { return this.storedNum; }

    public SetNum(int newVal)
    {this.storedNum = newVal; }

    public String toString()
    {return ""+this.storedNum; }
}