Answer questions 1 to 7 on a word file; write a program for each of Q.8 - Q.10.

1. (13 points)Give a brief answer for each of the following questions:
2. What is the difference between an interpreted language and a compiled language?
   1. **An interpreter will read each statement from the source code and translate it to machine code and execute it right away. A compiled language will translate the entire source code to machine code which can then be executed when the user pleases.**
3. What does JDK stand for? What does JRE stand for?
   1. **JDK(Java Development Kit): a set of programs which are invoked from the command line, for compiling, running, and testing Java programs.**
   2. **JRE(Java Runtime Environment): the program which runs the java programs.**
4. What is the Java source filename extension, and what is the Java bytecode filename extension?
   1. **The Java source filename extension is .java, and the bytecode filename is .class .**
5. What is the command to compile a Java program?
   1. **javac [program-name.java]**
6. What is the command to run a Java program?
   1. **java [program-name]**
7. Explain the two compilation phases of Java programs.
   1. **During the first phase java code is compiled into bytecode. In the second phase bytecode is read and interpreted into machine code.**
8. Show the output of the following code:

**double** amount = 5;

System.out.println(amount / 2);

System.out.println(5 / 2);

**OUTPUT:**

**2.5**

**2**

1. What data types are required for a switch variable? If the keyword break is not used after a case is processed, what is the next statement to be executed?
   1. **The data types required for a switch variable are char, byte, short, or int.**
   2. **If the keyword break is not used the next case statement will be executed.**
2. What is y after the following switch statement is executed? Rewrite the code using an if-else statement.

x = 3; y = 3;

**switch** (x + 3) {

**case** 6: y = 1;

default: y += 1;

}

* 1. **After the above switch statement y is equal to 2.**
  2. **Rewrite the code**

**x = 3; y = 3;**

**if(x + 3 == 6) {**

**y = 1;**

**}**

**else {**

**y += 1;**

**}**

1. Why does the Math class not need to be imported?
   1. **The Math class is part of the java.lang package and is therefore already imported.**
2. Which of the following are correct literals for characters?

'1', '\u345dE', '\u3fFa', '\b', '\t'

* 1. **‘1’ yes is a correct character literal it stands for 1.**
  2. **‘\u345dE’ no this is not correct it has too many characters.**
  3. **‘\u3fFa’ yes this is a correct character literal**
  4. **‘\b’ yes is a correct literal it stands for backspace**
  5. **‘\t’ yes is a correct literal it stands for tab**

1. Write the code that generates a random lowercase letter.

**public class RandomLetter {**

**public class static void main(String[] args) {**

**int dec = 97 + (int)(Math.random() \* 26);**

**char ch = (char)dec;**

**System.out.println(ch);**

**}**

**}**

1. What is wrong in the following code?

**import** java.util.Scanner;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.in);

System.out.print("Enter an integer: ");

**int** value = input.nextInt();

System.out.println("The value is " + value);

System.out.print("Enter a line: ");

String line = input.nextLine();

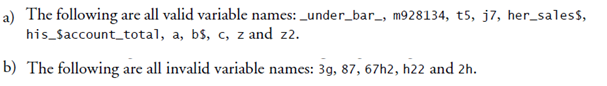
System.out.println("The line is " + line);

}

}

1. **input.nextline(); needs to be placed before the System.out.print(“Enter a line: “); because the \n must be removed before the user can input more characters.**

1. (2 points) State whether each of the following is true or false. If false, explain why.



1. **True all are valid variable names.**
2. **False, only h22 is valid because it starts the variable name must always start with a letter, $, or an underscore.**
3. (2 points) Write an expression that obtains a random integer between 34 and 55 inclusive. Write an expression that obtains a random integer between 0 and 999.

**int randInt = 34 + (int)(Math.random() \* 22);**

1. (2 points) Can the following conversions involving casting be allowed? If so, find the converted result.

**char** c = 'A';

**int** i = (**int**)c;

**float** f = 1000.34f;

**int** i = (**int**)f;

**double** d = 1000.34;

**int** i = (**int**)d;

**int** i = 97;

**char** c = (**char**)i;

**Yes they can the results will be as follows**

1. **65**
2. **1000**
3. **1000**
4. **a**
5. (5 points) Suppose that s1, s2, and s3 are three strings, given as follows:

String s1 = "Welcome to Java";

String s2 = "Programming is fun";

String s3 = "Welcome to Java";

What are the results of the following expressions?

1. s1 == s2
   1. **False**
2. s1 == s3
   1. **True**
3. s1.equals(s2)
   1. **False**
4. s1.equals(s3)
   1. **True**
5. s1.compareTo(s2)
   1. **7**
6. s2.compareTo(s3)
   1. **-7**
7. s2.compareTo(s2)
   1. **0**
8. s1.charAt(0)
   1. **W**
9. s1.length()
   1. **15**
10. s1.toLowerCase()
    1. **welcome to java**
11. (4 points) Suppose that s1 and s2 are two strings. Which of the following statements or expressions are incorrect?

String s1 = "Welcome to Java";

String s2 = "Welcome to Java";

String s3 = s1 + s2;

String s3 = s1 - s2;

s1 == s2;

s1 >= s2;

s1.compareTo(s2);

**char** c = s1(0);

**char** c = s1.charAt(s1.length());

1. **String s3 = s1 – s2; Wrong cannot subtract a string from a string.**
2. **s1 >= s2; Wrong, cannot have a string greater than another string.**
3. **char c = s1(0); Wrong need s1.charAt(0) to get first char in string.**
4. **char c = s1.charAt(s1.length()); Wrong, s1.length() is outside the range of the string.**
5. (4 points) Show the output of the following statements.

(a) System.out.printf("amount is %f\n", 32.32);

i. **amount is 32.320000**

(b) System.out.printf("amount is %5.2f%\n", 32.327);

i. **Exception**

(c) System.out.printf("%6b\n", (1 > 2));

i. **false**

(d) System.out.printf("%6s\n", "Java");

i. **Java**

(e) System.out.printf("%-6b%s\n", (1 > 2), "Java");

i. **false Java**

(f) System.out.printf("%6b%-8s\n", (1 > 2), "Java");

i. **falseJava**

(g) System.out.printf("%,5d %,6.1f\n", 312342, 315562.932);

i. **312,342 315,562.9**

(h) System.out.printf("%05d %06.1f\n", 32, 32.32);

i. **00032 0032.3**