Homework 5 Report

Q1: A do-while statement or **do-while loop** is very similar to the while statement. The main difference is that body of a do-while loop always executed at least once.

For statement or **for loop** enables to write easily a loop that controlled by some sort of counter. This loop used to iterate a part of the program several times. If the number of iteration fixed it is recommended. Traditional for statement represented as below.

```
for ( Initializing_Action ; Boolean_expression ; Update_Action )

          {
            Statements
          }
```

There is another form of the for statement is used when we have a collection of data such as an enumeration. This is called **for-each** statement. It generally used traversing array or collection in java. We don’t need to updating the increment value. It works on element basis.

A **while** statement repeats its action repeatedly as long as a controlling boolean expressions is true The loop is repeated while the controlling boolean expression is true.

In question, **while** loop is known for boolean condition and enter the inner statements.

Answer is Option D

Q2: According the explanation of loop types **for** loop is known controlling by a counter in previous answer.

Answer is Option B

Q3: A do-while statement or **do-while loop** is very similar to the while statement. The main difference is that body of a do-while loop always executed at least once.

For statement or **for loop** enables to write easily a loop that controlled by some sort of counter. This loop used to iterate a part of the program several times. If the number of iteration fixed it is recommended. Traditional for statement represented as below.

```
for ( Initializing_Action ; Boolean_expression ; Update_Action )

          {
            Statements
          }
```

There is another form of the for statement is used when we have a collection of data such as an enumeration. This is called **for-each** statement. It generally used traversing array or collection in java. We don’t need to updating the increment value. It works on element basis.

A **while** statement repeats its action repeatedly as long as a controlling boolean expressions is true The loop is repeated while the controlling boolean expression is true.

In this question asks about the execute at least one. This is lead us to the **do-while loop.**

Answer is Option A

Q4: According the explanation of loop types **for-each** loop is known referring with elements in previous answer.

Answer is Option C

Q5: **Break:** The keyword used for terminate the loop or a switch case statements.

Continue: Execution of a program ends in current state inner loop and contınues to the next iteration of a loop.

There is not any keywords or explanations for **end** and **skip** in Java.

Answer is Option B

Q6: According to the explanations in previous answer break keyword used for ends the execution flow and proceeded it the after loop statements.

Answer is Option A

Q7: **do while** loop has do keyword and braces added for statements. After the ending brace while keyword used with boolean expression in parenthesis.

```
do {
      Statements
   } while (Boolean_expression)
```

for loop has three expression in parenthesis. At first there is an initialize the counter variable then this literal checked with the boolean expression. Iterate operation proceeded by the counter variable in last section.

```
for ( Initializing ; Boolean_expression ; increment )
```

foreach used for some action for each item in an enumeration. First expression represents the type and object created by a programmer. After that declared the where the object traverse in an collection or may be a array. These are separated with a colon.

```
for (Card nextCard : Card.values())
```

while loop include a boolean expression in parenthesis. After that the statement located between braces.

```
while ( Boolean_expression )
{
  Statements
}
```

This explanations show us the only for statement has three segments.

Answer is Option B

Q8: The **for** statement it is used for iterating to the end of index.

```
for (int i = 0 ; i < arraySize(); i++ )
```

Also same statement is used for array last index.

```
for (int i = array.length()-1 ; i>=0 ; i-- )
```

Finally, both of the statements are true.

Answer is Option C

Q9: The usage of for-each loop it does not allow to modify list. We don’t know the index of the element. It works on elements basis not an index. So we can not mention about index 0 . Thus, we cannot mention about the last index. For-each loop always start form first element of a collection.

Answer is Option A

Q10: The loop body is executed even though the boolean expression starts out false in do while loop. İt checks the boolean value after the one iteration in the loop.

Answer is Option A

Q11: While statement accept the boolean expression to working clearly. The integer variable singer can type mismatching situation in the loop. Therefore, the code does not compile.

Answer is Option B

Q12: There is a defined list consist of string data type. The list has two element can and cup. For statement is start from last index of array to the first element. The code is print out from the last element **“cup”** separate with a comma (,) and print the element **“can”** add the comma again.

Answer is Option B

Q13: In this question for loop written with absent of increment/decrement operator. This cause the for loop executing without increment or decrement operation. When the loop iterates one time the first element **“glass”** printed. After that, the loop terminate with **break** statement. The loop not going through again. At the final step is print the **“end”** word to the screen. If we remove the break keyword in this code, it entered the infinite loop. Only prints the glass word numbers of times.

Answer is Option A

Q14: The following code create a letter reference for empty string. If only one statement there, no need to write the curly braces. Then execution flow enter the while loop adds **“a”** two times. Because **while loop** is iterated two times. Moreover, the code prints the **“aa”** to the screen.

Answer is Option A

Q15: The code is executed with September 3 1940 arguments. These values assigned to the string array and it has three elements. Args.length expression equals the three. Program flow enters **for** loop and args.length always bigger than zero (3 > 0). Furthermore, the index is incremented. So there is an infinite loop occurred. This code always print the args word. This code does not behave in the options. The problem can solved with change of the decrement operator rather than increment operator.

Answer is Option D

Q16: The following code there is a string array called stops that has four elements. Count variable not initialized and has a default value zero. Count smaller than the arrays length (0 < 4) so program entered the while loop. Stop array at zero index and looking for the first word’s length. First element is **Washington** and it has ten character. Ten bigger than the eight and the program flow can increase the count (1) and stop index looking for **Monroe**. This word has six characters end entered the if statement. The flow across the break statement and exit from if statement. By the way count variable already increased to two . So the result is two.

Answer is Option B

Q17: The code is does not compile because count variable defined at the first **do-while** loop and its scope cover the only outer do while loop braces. So count is not defined the out of the loop and take cannot resolve a variable problem error.

Answer is Option C

Q18: In for loop incremental operator can negligible. Because if we enter the **for** loop we can increment or decrement operation in the loop scope. Other that first and second segments are interdependent to each other.

Answer is Option C

Q19: **for (; ;)** and **while (true)** and **do{} while(true)** all these are candidate for infinite loops. Enhanced for used to traverse in array or collection so where the collection ends the loop is ends. There is not any probability for goes infinitive.

Answer is Option C

Q20: There is a defined list consist of string data type. The list has two element can and cup. For statement is start from first index of array to the last element. The code is print out from the first element **"can"** separate with a comma (,) and print the element **"cup"** add the comma again.

Answer is Option A

Q21: In **do-while** loops the body executed whatever happens. So code outputs helium once.
Answer is Option B

Q22: This code snippet returns the initial values of given string array. This enhanced for loop contains a colon and contains string fun. Then a new object traverse about fun array. The option B has the meet the condition in this question
Answer is Option B

Q23: Labeled break statement used to terminate a **for**, **while** or **do-while** loops. A labeled break terminates the outer loop. The control flow is transferred the statement of after the outer loop. In this question, there is an inner loop structure.

In this structure, If we use the only **break** statement, the inner loop (*starts with numbers:*) to terminate and the outer loop (*starts with letters:*) continues to the iterating.

If we use the **break letters** statement, the outer loop (*starts with letters:*) discarded and the inner loop (*starts with numbers:*) executed only once.

If we use **break numbers** statement, the outer loop (*starts with numbers:*) to terminate and the other loop (*starts with letters:*) iterates.

The result is **break numbers** and **break** statements code flow the beginning of letter loop.

Resource: https://www.baeldung.com/java-continue-and-break

Answer is Option C

Q24: Labeled continue statement used to terminate a **for**, **while** or **do-while** loops. A labeled break terminates the outer loop. The control flow is transferred the statement of after the outer loop. In this question, there is an inner loop structure.

In this structure, If we use the only **continue** statement, the inner loop and outer loop continues to the iterating.

If we use the **continue letters** statement, the outer loop (*starts with letters:*) iterated and the inner loop (*starts with numbers:*) executed only once.

If we use **continue numbers** statement, the outer loop (*starts with numbers:*) to iterate and the other loop (*starts with letters:*) iterates.

The result is continue numbers and break statements code flow the beginning of letter loop.

Answer is Option C

Q25: This code terminated without any output. Because the singer value equals the zero and don't enter the while loop.

Answer is Option C

Q26: Enhanced for loop provides to iterate through the elements of a collection or array. Arraylist and String builder can regarded collection. Int is a primitive data type also mean that it is not a class so taxis can not allowed with usage of int[ ].

Answer is Option B

Q27: In do while statement do section executed correctly. After the exit of if statement the outer boolean definition is vaild. Then the control flow print the output.

Answer is Option B

Q28: The following code enters the infinite loop. Becasue the letter string always empty. The while loop always added the string.

Answer is Option D

Q29: **for** loop has three expression in parenthesis. At first, there is an initialize the counter variable then this literal checked with the **boolean** expression. Iterate operation proceeded by the counter variable in last section.

```
for ( Initializing ; Boolean_expression ; increment )
```

Answer is Option B

Q30: In given code char array size increasing and the count is decreased minus 1 and 2 and etc. At the end count equal four and break out from the while loop.

Answer is Option B

Q31: The i variable equals to ten. After that the control flow enter the while loop. Then i variable iterated three times. Other that k is incremented once. The result is one.

Answer is Option A

Q32: This code snippet returns the initial values of given string array. This enhanced for loop contains a colon and contains String fun. Then a new object traverse about fun array. The option B has the meet the condition in this question

Answer is Option B

Q33: In this question for loop written with absent of curly braces. The borders of for loop not assigned and break stays outside. Then this code gives the compile error.

Answer is Option C

Q34: The does not compile because there is semicolon near the i++ operator. It need to change with a comma.
Answer is Option C

Q35: The code is executed with September 3 1940 arguments. These values assigned to the string array and it has three elements. Args.length expression equals the three. Program flow enters **for** loop and args.length always bigger than zero (3 > 0). Furthermore, the index is incremented. So there is an exception (*ArrayIndexOutOfBoundsException*) occurred. There is not a arg[3] element .

Answer is Option C

Q36: In the code there is an initialization performed, the String tie the result is shoelace

Answer is Option B

Q37: If we remove the two of lines, the break statement stays an outside and it gives the compile error.

Answer is Option C

Q38: The following code there is a string array called stops that has four elements. Count variable not initialized and has a default value zero. Count smaller than the arrays length (0 < 4) so program entered the while loop. Stop array at zero index and looking for the first word's length. First element is **Washington** and it has ten character. Ten bigger than the eight and the program flow can increase the count (1) and stop index looking for **Monroe**. This word has six characters end entered the **if** statement. The flow across the **continue** statement and iterate the statement. By the way, count variable already increased to two. The count incremented till the arrays end. Therefore, the result is four.

Answer is Option C

Q39: In this code there is a **boolean** value in while loop. Builder not a **boolean** value, so the code gives the compile error.

Answer is Option C

Q40: Control flow enter the inner **do while** loop. It increased the two. The flow breaks out the loop the outer loop always true. Then break ends the outer loop. Finally, it prints two the screen.

Answer is Option A

Q41: There is not any option for code compiles correctly and not enter the infinite loop.
Answer is Option D

Q42: There are two-string array in this code. The **for** statement tries to merge them and separate with them the minus operator. The result is Downtown Day-Uptown Night-
Answer is Option B

Q43: This code compile correctly it print all the possible pairs and it prints four lines.

Answer is Option B

Q44: If there is a false boolean expression handled for statement only alpha is executed. The others neglected.

Answer is Option A

Q45: If there is a loop body runs in once looks the alpha and beta firstly. Then the control follow enter the statement. Once the executed then it iterated by gamma then checked again with beta.

Answer is Option B

Q46: A option is iterated five times, B option is iterated five times, C option is iterated five times because of the do statement but Option D is iterated four times.

Answer is Option D

Q47: In this code infinite loop because the while statement ended with a semicolon. IT returns always true.

Answer is Option D

Q48: In this code trying to do label break statement but. For is used in Java programs. It cannot allowed the for keyword such like that. The code give the compile error.

Answer is Option C

Q49: The code compiled control flow enter the **if** statement this is true and prints the inflate- to the screen. The balloonInflated defined at the outside the loop so it can class variable. Moreover, it cannot access the while loop and there is one word in output.

Answer is Option D

Q50: The blanks is first segment of for loop. This section is generally initialized some variables. There are two variables needed and start from zero. And these are separated by a comma (,). All these ideas meets the option B.

Answer is Option B