Loops - 2

- A general while loop has the following structure:

```
while (condition) {

    // code to execute

    // update condition
}
```

- A sentinel loop is a special type of while loop that continues to execute until a special value is entered. For example, the following code will continue to prompt the user for a number until they enter -1.

```
Scanner scanner = new Scanner(System.in);
int number = 0;
while (number != -1) {
    System.out.print("Enter a number (-1 to quit): ");
    number = scanner.nextInt();
}
```

- do while loops are similar to while loops, except that the condition is checked at the end of the loop instead of the beginning. This means that the loop will always execute at least once.

```
do {
    // code to execute
} while (condition);
```

- for loops are a special type of loop that are used when you know how many times you want to execute the loop. The general structure of a for loop is:

```
for (initialization; condition; update) {
    // code to execute
}
```

- `break` is a keyword that can be used to exit a loop (or any other block of code). For example, the following code will continue to prompt the user for a number until they enter -1, at which point the loop will exit.

```
import java.util.Scanner;

public class punchcard {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        while (true) {
            System.out.print("enter a number (-1 to quit): ");
            int number = scanner.nextInt();
            if (number == -1) {
                break;
            }
        }
        System.out.println("goodbye!");
    }
}
```

- `continue` is a keyword that can be used to skip the rest of the current iteration of a loop. For example, the following code will continue to count from 1 to 10 but will skip the number 5.

```java
public class punchcard {
    public static void main(String[] args) {
        int count = 0;
        while (count < 10) {
            count++;
            if (count == 5) {
                continue;
            }
            System.out.println(count);
        }
    }
}
```

1. Fill the blank so the while loop will only execute 10 times.

```
public class punchcard {
    public static void main(String[] args) {
        int i = 1;
        while (______) {
            System.out.println("entered loop " + i + " times");
            i++;
        }
    }
}
```

a) `i < 9`
b) `i <= 9`
c) `i < 10`
d) `i <= 10`
e) `i <= 11`

answer: d - `i <= 10`

---

2. how many times will the following loop execute?

```
public class punchcard {
    public static void main(String[] args) {
        boolean flag = false;
        while (flag) {
            System.out.println("in the loop body");
        }
        System.out.println("after the loop");
    }
}
```

a) 0
b) 1
c) Infinite
d) compile error
e) runtime error

answer: a - 0

---

3. how many times the following loop execute?

```
public class punchcard {
    public static void main(String[] args) {
        do {
            System.out.println("in the loop body");
        } while (false);
        System.out.println("after the loop");
    }
}
```

a) 0
b) 1
c) Infinite
d) compile error
e) runtime error

answer: b - 1

---

4. Fill the blank so the for loop will only execute 5 times.

```java
public class punchcard {
    public static void main(String[] args) {
        int i = 0;
        do {
            System.out.println("in the loop body");
            ---------- {
                break;
            }
            i++;
        } while (true);
        System.out.println("after the loop");
    }
}
```

a) `if (i <= 4)`
b) `if (i == 4)`
c) `while (i < 4)`
d) `while (i <= 4)`
e) `else if (i <= 4)`

answer: b - `if (i == 4)`

---

5. how many times will the following loop execute?

```java
public class punchcard {
    public static void main(String[] args) {
        for (int i = 0; i < 10; i=i+2) {
            System.out.println("in the loop body");
        }
        System.out.println("after the loop");
    }
}
```

a) 0
b) 1
c) 5
d) 10
e) Infinite

answer: c - 5

---

6. how many times will the following loop execute?

```java
public class punchcard {
    public static void main(String[] args) {
        boolean flag = true;
        for (int i = 0; flag; i++) {
            System.out.println("in the loop body for the " + i + "th time");
            if (i*i == 64) {
                flag = false;
            }
        }
    }
}
```

a) 0
b) 1
c) 4
d) 8

e) 9

---

7. I'm trying to make a menu system for a accounting software. I want to keep asking the user for a menu option until they enter 0 to quit. What is wrong with the following code?

```
import java.util.Scanner;

public class punchcard {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        do {
            int number = scanner.nextInt();
            switch(number) {
                case 1:
                    System.out.println("Language selection");
                    break;
                case 2:
                    System.out.println("Customer support");
                    break;
                case 3:
                    System.out.println("Check the balance");
                    break;
                case 4:
                    System.out.println("Check loan balance");
                    break;
            }
        } while(number != 0);
        System.out.println("Exit");
    }
}
```

a) The `scanner` object is not properly initialized.
b) There is no default case in the switch statement.
c) The `number` variable is not declared before the do-while loop.
d) The loop condition doesn't allow the user to exit when entering 0.
e) The code should function as intended. Nothing is wrong.

answer: c - The `number` variable is not declared before the do-while loop.

---

8. what will be the output of the following code when executed?

```
public class punchcard  {
    public static void main(String[] args) {
        String message = "Hello, World!";
        for (int i = 1; i < message.length(); i = i*2) {
            char currentChar = message.charAt(i);
            if (Character.isLetter(currentChar)) {
                System.out.print(Character.toUpperCase(currentChar));
            } else {
                System.out.print(currentChar);
            }
        }
    }
}
```

a) Hello, World!
b) hELLO, wORLD!
c) H, W!
d) HELLOWORLD

e) ELOO

answer: e - ELOO

---

9. What does the following code do?

```
public class punchcard  {
    public static void main(String[] args) {
        String input = "Java Programming";
        String reversed = "";

        for (int i = input.length() - 1; i >= 0; i--) {
            reversed += input.charAt(i);
        }

        System.out.println(reversed);
    }
}
```

a) Prints the original string without any changes.
b) Generates an error due to incorrect loop implementation.
c) Reverses the characters in the string and prints the result.
d) Removes all vowels from the string.
e) Prints the string in all uppercase letters.

answer: c - Reverses the characters in the string and prints the result.

---

10. What will be the output of the following code when executed?

```
public class StringModification {
    public static void main(String[] args) {
        String text = "Welcome to Java";
        String modifiedText = "";

        for (int i = 0; i < text.length(); i++) {
            char currentChar = text.charAt(i);
            if (currentChar != ' ') {
                modifiedText += Character.toUpperCase(currentChar);
            } else {
                modifiedText += "\b";
            }
        }

        System.out.println(modifiedText);
    }
}
```

a) WELCOME TO JAVA
b) WelcomeToJava
c) Welcome to Java
d) welcomeTOjava
e) WELCOM T JAVA

answer: e - WELCOM T JAVA

---

11. What will be the output of the following code when executed?

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

```
        String sentence = "Java fun. java fun. JavaFun. Jav fun.";
        int count = 0;

        for (int i = 0; i < sentence.length() - 3; i++) {
            // if (sentence.substring(i, i + 4).equals("Java")) {
            //  count++;
            // }
            int index = sentence.indexOf("Java", i);
            if (index != -1) {
                count++;
                i = index;
            }
        }

        System.out.println("The word 'Java' appears " + count + " times.");
    }
}
```

a) 1
b) 2
c) 3
d) 4
e) error

answer: b - 2

---

12. What should be the termination condition of the following loop?

```
public class punchcard {
    public static void main(String[] args) {
        String string = "ZERO-ONEE-ZERO-ZERO-ONEE-ZERO-ZERO-ZERO-ZERO-ZERO-ONEE-ONEE-ZERO-ONEE-ZERO-ZERO-ZERO-ON
        int index = 0;
        while (________________________) {
            int nextHyphen = string.indexOf("-", index);
            if (nextHyphen == -1) {
                nextHyphen = string.length();
            }
            String number = string.substring(index, nextHyphen);
            switch (number) {
                case "ZERO":
                    System.out.print("0");
                    break;
                case "ONEE":
                    System.out.print("1");
                    break;
            }
            index = nextHyphen + 1;
        }
    }
}
```

a) `index < string.length()`
b) `index < string.indexOf("-")`
c) `index < string.indexOf("-", i)`
d) `index < string.length() - 4`
e) `index < string.indexOf("-ONEE\n")`

answer: d - `i < string.length() - 4`

NOTE: using string.split() would be a better solution to this problem. but since we haven't covered that yet, we'll use the above

solution.

---

13. The output of the privious code is still meaningless. to advance with the decryption, we need to put an space between every 8 bits.

what should be in the blank so the code snippet will insert a space after every 8 bits?

```
public class PunchCard {
    public static void main(String[] args) {
        String string = "01001000001101000100001101001011001100110101001000110101";
        int i;
        for (i = 0; _____________________________) {
            System.out.print(string.substring(i, i + 8) + " ");
        }
    }
}
```

a) `i + 8 <= string.length(); i += 8`
b) `i < string.length(); i ++`
c) `i + 8 <= string.length(); i ++`
d) `i > string.length(); i += 8`
e) `i == string.length(); i ++`

answer: a - `i + 8 <= string.length(); i += 8`

To translate the binary code into decimal, we need to use the following code snippet:

since we haven't covered some off the methods used in the following code, we will use them without explanation.

```
public class BinaryToDecimal {
    public static void main(String[] args) {
        String binaryString = "01001000 00110100 01000011 01001011 00110011 01010010 00110101";

        // Split the binary string into separate binary numbers
        String[] binaryNumbers = binaryString.split(" ");

        // Convert each binary string to decimal and print
        for (String binary : binaryNumbers) {
            int decimal = Integer.parseInt(binary, 2);
            System.out.print(decimal + " ");
        }
    }
}
```

output:

```
72 52 67 75 51 82 53
```

---

14. Now we have the decimal numbers, we need to convert them to ASCII characters. What is wrong with the following code?

```
public class PunchCard {
    public static void main(String[] args) {
        String input = "72 52 67 75 51 82 53";

        for (i = 0; i < input.length(); i += 3) {
            System.out.print((char) Integer.parseInt(input.substring(i, i + 2)));
        }
    }
}
```

output:

```
H4CK3R5
```

a) The update statement of the for loop should be `i += 2`
b) The substring method should be using `i + 3` instead of `i + 2` (`input.substring(i, i + 2)`)
c) The iteration variable `i` should be declared.
d) the condition of the for loop should be `i < input.length() - 2` to stop at the last character.
e) nothing is wrong with the code.

answer: c - The iteration variable `i` should be declared.

---

15. we came pretty far with the decryption. but we still have a problem. no body knows the leet language, but we have of characters. we need to convert the leet characters to normal characters.

dictionary:

```
4 = A
8 = B
3 = E
6 = G
1 = I
0 = O
2 = Z
```

what will be the output of the following code when executed?

```java
public class PunchCard {
    public static void main(String[] args) {
        String input = "H4CK3R5";

        for (int i = 0; i < input.length(); i++) {
            char currentChar = input.charAt(i);
            switch (currentChar) {
                case '4':
                    System.out.print("A");
                    break;
                case '8':
                    System.out.print("B");
                    break;
                case '3':
                    System.out.print("E");
                    break;
                case '6':
                    System.out.print("G");
                    break;
                case '1':
                    System.out.print("I");
                    break;
                case '0':
                    System.out.print("O");
                    break;
                case '2':
                    System.out.print("Z");
                    break;
                default:
                    System.out.print(currentChar);
            }
        }
    }
}
```

a) H4CK3R5
b) H4CEKR5
c) HAGKEBZS
d) HZCK3R5
e) HACKER5

answer: e - HACKER5