

While loop

A general while loop has the following structure:

```
while (condition) {  
  
    // code to execute  
  
    // update condition  
}
```

The condition is a boolean expression that is evaluated at the beginning of each iteration of the loop. If the condition is true, the code inside the loop is executed. After the code is executed, the condition is evaluated again. If the condition is still true, the code is executed again. This process repeats until the condition is false.

The following example prints the numbers 1 through 10:

```
int i = 1;  
while (i <= 10) {  
    System.out.println(i);  
    i++;  
}
```

The variable `i` is initialized to 1. The condition `i <= 10` is evaluated. Since `i` is 1, the condition is true. The code inside the loop is executed, which prints the value of `i` and increments `i` by 1. The condition is evaluated again. Since `i` is now 2, the condition is true. The code inside the loop is executed again, which prints the value of `i` and increments `i` by 1. This process repeats until `i` is 11. When `i` is 11, the condition `i <= 10` is false, so the loop terminates.

Note: A while loop can be used inside any other loop or conditional statement and vice versa.

1. What's the output of the following code?

```
public class punchcard {  
    public static void main(String[] args) {  
        int x = 5;  
        while (x > 0) {  
            System.out.print(x + " ");  
            x -= 2;  
        }  
    }  
}
```

- A) 5 4 3 2 1
- B) 5 3 1
- C) 5 2
- D) 5 2 0
- E) 5 3 1 -1

answer: B - 5 3 1

2. Which options can fill in the blank, So the code prints the even numbers from 0 to 10?

Expect output:

```
Counting: 0  
Counting: 2  
Counting: 4  
Counting: 6  
Counting: 8
```

```
public class punchcard {  
    public static void main(String[] args) {  
        int i = 0;  
        while (_____) {  
            System.out.println("Counting: " + i);  
            _____;  
        }  
    }  
}
```

- A) (i < 10) ~~ i += 2
- B) (i <= 10) ~~ i += 2
- C) (i < 10) ~~ i ++
- D) (True) ~~ i += 2
- E) (i <= 10) ~~ i ++

answer: A - (i < 10) ~~ i += 2

3. Identify the Incorrect Part of the Code. Suppose the goal is to print the numbers 10 through 5 in descending order.

```
public class punchcard {  
    public static void main(String[] args) {  
        int num = 10;  
        while (num < 5) {  
            System.out.println(num);  
            num--;  
        }  
    }  
}
```

- A) The loop runs indefinitely.

- B) The decrement should be `num -= 2`;
- C) The loop will not execute at all.
- D) The decrement should be `num++`;
- E) The loop will execute only once.

answer: C

4. What's the correct output of the following code?

```
public class punchcard {  
    public static void main(String[] args) {  
        int y = 3;  
        while (y < 10) {  
            System.out.print(y * 2 + " ");  
            y += 3;  
        }  
    }  
}
```

- a) 6 9
- b) 6 12 15 18
- c) 6 12 18
- d) 6 9 12 15 18
- e) error

Answer: c) 6 12 18

5. Identify the Incorrect Part of the Code.

```
public class punchcard {  
    public static void main(String[] args) {  
        int z = 0;  
        while (z != 7) {  
            System.out.println(z);  
            z += 2;  
        }  
    }  
}
```

- a) The loop runs indefinitely.
- b) The loop body won't execute.
- c) The loop condition is incorrect.
- d) The loop body will execute once.
- e) There is no problem with the code.

Answer: a) The loop runs indefinitely.

6. Choose the Correct Answer to Fill the Missing Blank. The goal is to print the powers of 2 from 2 to 1024.

```
public class punchcard {  
    public static void main(String[] args) {  
        int e = 2;  
        while (e <= 1024) {  
            System.out.println("Value: " + e);  
            -----;  
        }  
    }  
}
```

- a) `e = e * e;`

- b) `e += 2;`
- c) `e = Math.pow(e, 2);`
- d) `e = Math.pow(2, e);`
- e) `e *= 2;`

Answer: e) `e *= 2;`

7. Identify the Incorrect Part of the Code.

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

        int f = 10;
        while (f != 0) {
            System.out.println(f);
            f -= 0;
        }
    }
}
```

- a) The loop runs indefinitely.
- b) The loop body won't execute.
- c) The loop condition is incorrect.
- d) The loop body will execute once.
- e) There is no problem with the code.

Answer: a) The loop runs indefinitely.

8. What's the output of the following code?

```
public class punchcard {
    public static void main(String[] args) {
        int g = 1;
        while (g <= 5) {
            System.out.print(g * g + " ");
            g++;
        }
    }
}
```

- a) 1 4 9
- b) 1 4 9 16 25
- c) 1 2 3 4 5
- d) 1 3 5 7 9
- e) 1 3 5 9 16 25

Answer: b) 1 4 9 16 25

9. What's the output of the following code?

```
public class punchcard {
    public static void main(String[] args) {
        int x = 10;
        while(x > 4) {
            System.out.print(x + " ");
            x--; //decrement by 1 here
            if (x == 5) {
                //when x reaches 5 increase value by 5, so the whole thing repeats again
                x += 5;
            }
        }
    }
}
```

```

    }
}
}
}

```

- a) 10 9 8 7 6 5 4 3 2 1
- b) 10 9 8 7 6 10 9 8 7 6 ...
- c) 10 9 8 7 6 5
- d) 10 11 12 13 14 ...
- e) error

Answer: b) 10 9 8 7 6 10 9 8 7 6 ...

10. Quick!!! defuse the bomb!!!

```

import java.util.Scanner;

public class punchcard {
    public static void main(String[] args){
        int time;
        Scanner timeScan=new Scanner(System.in);
        System.out.print("Set timer: ");
        time=timeScan.nextInt();
        while(time>=0){
            System.out.println(time);
            time--;
        }
        if (time==--1)
            System.out.println("BOOOOOOMMM");
        else
            System.out.println("Bomb defused!!!");
    }
}

```

- a) 123456436754
- b) 2.5
- C) 0
- d) -12
- e) 1

Answer: a) -12

12. fill in the blank. So the code prints the numbers that are divisible by 3 or the modulus of 10 is 3.

```

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner read = new Scanner(System.in);
        int number = read.nextInt();
        int n=1;
        while(n<=number){
            if(n%3==0 __ n%10==3) { // the blank is in this line
                System.out.println(n);
            }
            n++;
        }
    }
}

```

- a) ||
- b) &&
- c) |
- d) or
- e) +

Answer: a) ||

13. how many times the loop will run?

```
public class Main {  
    public static void main(String[] args) {  
        int number = 5;  
        int x = number/3;  
        int y = number/x;  
        while (y <= number){  
            if (y % 2 == 0){  
                System.out.println(y);  
            }  
            y = y + 1;  
        }  
    }  
}
```

- a) 0
- b) 1
- c) 2
- d) 11
- e) infinite

Answer: b) 1