

# ***Object-Oriented Programming***

Practice Week2

# What is wrong in the following code?

```
if (score >= 60) System.out.println("D");  
else if (score >= 70) System.out.println("C");  
else if (score >= 80) System.out.println("B");  
else if (score >= 90) System.out.println("A");  
else System.out.println("F");
```



## Question 2. Boolean

x	y	x    y	x && y
true	true	true	true
true	false	true	false
false	true	true	false
false	false	false	false



## Question 2. Boolean

**True = 1, False = 0**

x	y	x    y	x && y
1 true	1 true	true	true
1 true	0 false	true	false
0 false	1 true	true	false
0 false	0 false	false	false

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**`(true) && (3 > 4)`**

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**1      &&      0**

**[true] && [3 > 4]**



## Question 2. Boolean

**True = 1, False = 0**

x	y	x    y	x && y
true 1	true 1	true	true
true 1	false 0	true	false
false 0	true 1	true	false
false 0	false 0	false	false

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**$!(x > 0) \ \&\& \ (x > 0)$**



**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**0                      &&                      1**  
**!( $x > 0$ ) && ( $x > 0$ )**

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**$(x > 0) \parallel (x < 0)$**

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**1**  
 **$(x > 0)$**   **$\parallel$**  **0**  
 **$(x < 0)$**



## Question 2. Boolean

**True = 1, False = 0**

x	y	x    y	x && y
true 1	true 1	true	true 1
true 1	false 0	true	false
false 0	true 1	true	false
false 0	false 0	false 0	false

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**$(x \neq 0) \parallel (x == 0)$**

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**1**  
 **$(x \neq 0) \parallel (x == 0)$**

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**$(x \geq 0) \parallel (x < 0)$**

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

**1**  
 $(x \geq 0) \parallel (x < 0)$



**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

$$(x \neq 1) == !(x == 1)$$

**3.10.1 Assuming that  $x$  is 1, show the result of the following Boolean expressions:**

$$\begin{array}{c} 0 \\ (x \neq 1) \end{array} == \begin{array}{c} 0 \\ !(x == 1) \end{array}$$



### Question 3. Switch

```
System.out.println("switch(1)");
switch(1) {
    case 1:
        System.out.println("one");
        break;
    case 2:
        System.out.println("two");
        break;
    case 3:
        System.out.println("three");
        break;
    default:
        System.out.println("default");
        break;
}
```



### Question 3. Switch

```
System.out.println("switch(1)");  
switch(1) {  
    case 1:  
        System.out.println("one");  
        break;  
    case 2:  
        System.out.println("two");  
        break;  
    case 3:  
        System.out.println("three");  
        break;  
    default:  
        System.out.println("default");  
        break;  
}
```



```
switch(1)  
one
```



## Question 1. If-elseif

```
public class Ifelseif {  
    Park-Minjoo  
    public static void main(String[] args){  
        Scanner scan = new Scanner(System.in);  
        double score = scan.nextDouble();  
  
        if (score >= 90) System.out.println("A");  
        else if (score >= 80) System.out.println("B");  
        else if (score >= 70) System.out.println("C");  
        else if (score >= 60) System.out.println("D");  
        else System.out.println("F");  
    }  
}
```



## Question 2. Switch

```
switch (date) {  
    case 0: numberStr = "Sunday"; break;  
    case 1: numberStr = "Monday"; break;  
    case 2: numberStr = "Tuesday"; break;  
    case 3: numberStr = "Wednesday"; break;  
    case 4: numberStr = "Thursday"; break;  
    case 5: numberStr = "Friday"; break;  
    case 6: numberStr = "Saturday"; break;  
    default: numberStr = "Not a valid day"; break;  
}
```



### Question 3. Day Of the Week

Enter today's day: 1

Enter the number of days elapsed since today: 3

Today is Monday and the future day is Thursday

**1 = Monday**

**1 (date) + 3(elapsed) = 4**

**4 % 7 = 4**

**4 = Thursday**

### Question 3. Day Of the Week

Enter today's day: 1

Enter the number of days elapsed since today: 3

Today is Monday and the future day is Thursday

```
int futureDate = (date + elapsed) % 7;
String dayoftheWeek = "";

switch(date){
    case 0: dayoftheWeek = "Sunday"; break;
    case 1: dayoftheWeek = "Monday"; break;
    case 2: dayoftheWeek = "Tuesday"; break;
    case 3: dayoftheWeek = "Wednesday"; break;
    case 4: dayoftheWeek = "Thursday"; break;
    case 5: dayoftheWeek = "Friday"; break;
    case 6: dayoftheWeek = "Saturday"; break;
}
```

**1 = Monday**  
**1 (date) + 3(elapsed) = 4**  
**4 % 7 = 4**  
**4 = Thursday**





### Question 3. Day of the Week

Enter today's day: 1

Enter the number of days elapsed since today: 3

Today is Monday and the future day is Thursday

```
if (futureDate == 0){  
    System.out.printf("Today's is %s and the future day is Sunday", dayoftheWeek);  
}else if(futureDate == 1){  
    System.out.printf("Today's is %s and the future day is Monday", dayoftheWeek);  
}else if(futureDate == 2){  
    System.out.printf("Today's is %s and the future day is Tuesday", dayoftheWeek);  
}else if(futureDate == 3){  
    System.out.printf("Today's is %s and the future day is Wednesday", dayoftheWeek);  
}else if(futureDate == 4){  
    System.out.printf("Today's is %s and the future day is Thursday", dayoftheWeek);  
}else if(futureDate == 5){  
    System.out.printf("Today's is %s and the future day is Friday", dayoftheWeek);  
}else if(futureDate == 6) {  
    System.out.printf("Today's is %s and the future day is Saturday", dayoftheWeek);  
}
```

#### Question 4. Palindrome

Enter a three-digit integer: 121

Enter

121 is a palindrome

```
// Test for palindrome
```

```
int digit1 = 
```

-> 1 (ones place)

```
int remaining = 
```

-> 2 (tens place)

```
int digit3 = 
```

-> 1 (hundreds place)

#### Question 4. Palindrome

Enter a three-digit integer: 121

Enter

121 is a palindrome

```
// Test for palindrome
```

```
int digit1 = (int)(number / 100);
```

→ 1 (ones place)

```
int remaining = number % 100;
```

→ 2 (tens place)

```
int digit3 = (int)(remaining % 10);
```

→ 1 (hundreds place)



#### Question 4. Palindrome

```
// Test for palindrome
```

```
int digit1 = (int)(number / 100);
```

**-> 1 (ones place)**

```
int remaining = number % 100;
```

**-> 2 (tens place)**

```
int digit3 = (int)(remaining % 10);
```

**-> 1 (hundreds place)**

```
// Display result
```

```
System.out.println(
```

```
    number + (  ? " is a " : " is not a ") + "palindrome");
```



## Question 4. Palindrome

```
// Test for palindrome
int digit1 = (int)(number / 100);
int remaining = number % 100;
int digit3 = (int)(remaining % 10);
```

**1 (hundreds place)**  
**==**  
**1 (ones place)**



```
// Display result
System.out.println(
    number + (                      ? " is a " : " is not a ") + "palindrome");
```

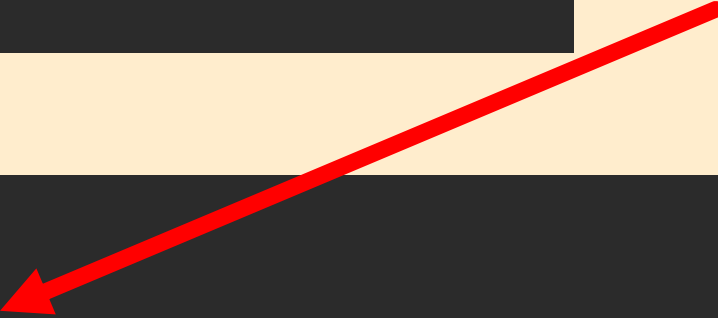


## Question 4. Palindrome

```
// Test for palindrome
int digit1 = (int)(number / 100);
int remaining = number % 100;
int digit3 = (int)(remaining % 10);
```

**1 (hundreds place)**  
**==**  
**1 (ones place)**

```
// Display result
System.out.println(
    number + ((digit1 == digit3) ? " is a " : " is not a ") + "palindrome");
```



### Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = ;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = ;

// Compute the current minute in the hour
long currentMinute = ;

// Obtain the total hours
long totalHours = ;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```

**1 sec = 1000 millis**

### Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = ;

// Compute the current minute in the hour
long currentMinute = ;

// Obtain the total hours
long totalHours = ;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```

**1 sec = 1000 millis**



### Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = ;

// Compute the current minute in the hour
long currentMinute = ;

// Obtain the total hours
long totalHours = ;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```

**Current = total % 60**

### Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = ;

// Compute the current minute in the hour
long currentMinute = ;

// Obtain the total hours
long totalHours = ;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```

**1 min = 60 s**

### Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = totalSeconds / 60;

// Compute the current minute in the hour
long currentMinute = ;

// Obtain the total hours
long totalHours = ;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```

**1 min = 60 s**





## Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = totalSeconds / 60;

// Compute the current minute in the hour
long currentMinute = ;

// Obtain the total hours
long totalHours = ;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```



### Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = totalSeconds / 60;

// Compute the current minute in the hour
long currentMinute = totalMinutes % 60;

// Obtain the total hours
long totalHours =                     ;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```

**1 hour = 60 m**

### Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = totalSeconds / 60;

// Compute the current minute in the hour
long currentMinute = totalMinutes % 60;

// Obtain the total hours
long totalHours = totalMinutes / 60;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```

**1 hour = 60 m**

### Question 5. 12-hour clock

```
// Obtain the total seconds since midnight, Jan 1, 1970
long totalSeconds = totalMilliseconds / 1000;

// Compute the current second in the minute in the hour
long currentSecond = totalSeconds % 60;

// Obtain the total minutes
long totalMinutes = totalSeconds / 60;

// Compute the current minute in the hour
long currentMinute = totalMinutes % 60;

// Obtain the total hours
long totalHours = totalMinutes / 60;

// Compute the current hour
long currentHour = totalHours % 24;
currentHour = currentHour + offset;
```

**1 hour = 60 m**



### Question 5. 12-hour clock

```
// Display results
System.out.println(
    "Current time is " + ((currentHour > 12) ? currentHour - 12 :
        currentHour) + ":" + currentMinute + ":" + currentSecond +
        ((currentHour > 12) ? " PM" : " AM"));
```

**If (currentHour) > 12**

**else If (currentHour) <= 12**



## Question 3. Stdin and Stdout

# https://www.hackerrank.com/

The screenshot shows the HackerRank website with a navigation bar at the top containing links for Products, Solutions, Resources, Pricing, For Candidates, Request Demo, and a Get Started button. The main content is split into two columns. The left column, titled 'For Companies' with a 'BUSINESS' tag, describes the platform as a market-leading technical interview tool and includes a 'Login' button and a link to 'Contact sales or Get free trial.' The right column, titled 'For Developers', invites over 21 million developers to practice coding and includes a 'Login' button. A '2023 Developer Skills Report' pop-up is visible on the right, offering to discover in-demand skills and languages. The footer features logos for Peloton, Atlassian, Bloomberg, VMware, Stripe, Goldman Sachs, Adobe, and LinkedIn, along with a cookie policy notice.

HackerRank

Products Solutions Resources Pricing For Candidates | Request Demo Get Started

**BUSINESS**

### For Companies

We are the market-leading technical interview platform to identify and hire developers with the right skills.

Login

Don't have an account?  
Contact sales or Get free trial.

### For Developers

Join over 21 million developers, practice coding skills, prepare for interviews, and get hired.

Login

Don't have an account?  
Sign up.

#### 2023 Developer Skills Report

Discover which developer skills and programming languages are the most in demand in 2023.

View Report

PELOTON ATLASSIAN Bloomberg vmware stripe Goldman Sachs Adobe LinkedIn

We use cookies to ensure you have the best browsing experience on our website. Please read our cookie policy for more information about how we use cookies. Find out more. OK

## Question 3. Stdin and Stdout

The screenshot displays the HackerRank 'Prepare' interface. At the top, the navigation bar includes 'HackerRank', 'PREPARE' (highlighted with a 'NEW' badge), 'CERTIFY', and 'COMPETE'. A search bar and user profile 'Mandy5G' are on the right. The main section is titled 'Prepare' with a 'Bookmarked Challenges' link. Under 'Your Preparation', there are two topic cards: 'Java' and 'Problem Solving'. Each card shows a progress bar, a percentage, and a 'Continue Preparation' button.

Topic	Progress	Next Step
Java	12% (Get to 25 points to unlock this badge)	Continue Preparation
Problem Solving	30% (49 points to next star)	Continue Preparation

<https://www.hackerrank.com/>

# *Introduction to Java*

*The term Java from Java island...*

*JIU IT Major 20230124 PMJ, AEJ*