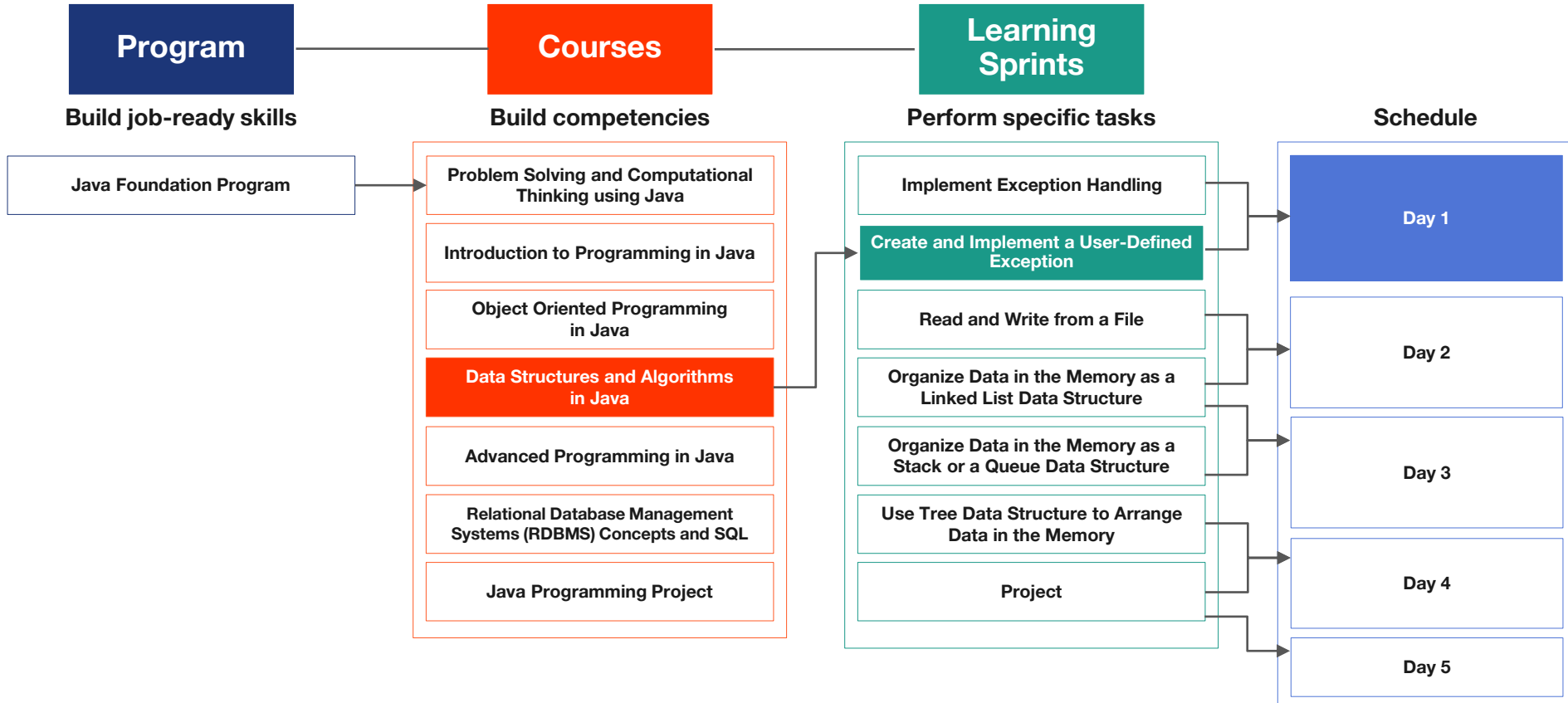


Java Program: Course 4: Plan

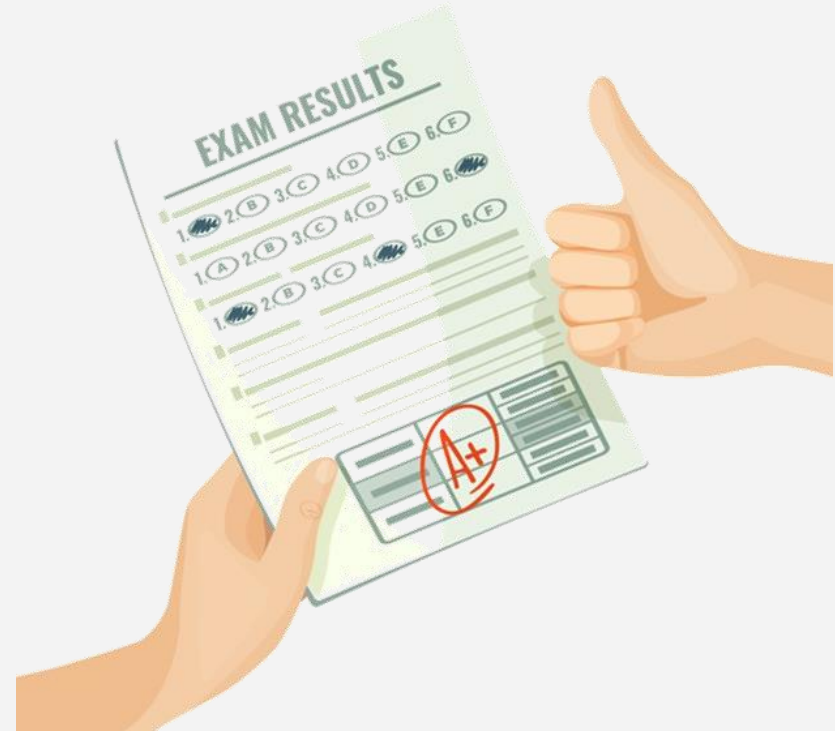


Automated Grading

Teachers, along with the technical team of Oak Bridge School have to automate their grading process. The teams have to:

- Accept the marks scored by each student in each subject
- Calculate the total marks scored by each student
- Evaluate the grades obtained by each student

If you were a part of their team, how would you ensure that the marks obtained do not exceed 100 in any of the subjects?



Opening Bank Accounts Online

The software team of IDBC Bank needs to build an application that will help customers to open a save and/or a pay bank account online.

As a member of their team, how will you ensure that the account balance is not a negative number?



Think and Tell

- How can we ensure the marks of a student in any of the subjects do not exceed 100 or that the account balance is not negative?
- Is there a predefined class that can be used as an exception handler here?
- What will you do if there is no suitable predefined class in the Java exception handling library?



Create and Implement a User-Defined Exception



Learning Objectives

- Explain user-defined exceptions
- Use the throws and throw keywords
- Implement user-defined exceptions
- Describe exception propagation



User-Defined Exceptions

- Java allows programmers to create their own exceptions
- Such exceptions are known as custom exceptions or user-defined exceptions
- User-defined exceptions are classes that provide flexibility to add attributes and methods that are not part of a standard Java exception library

Creating User-Defined Exceptions

- To create a user-defined exception class, the class needs to inherit the `java.lang.Exception` class, as shown below:

```
class AgeException extends Exception{  
  
}
```

- All user-defined exceptions are checked exceptions

Declaring User-Defined Exceptions

If a method does not handle a checked exception, the `throws` keyword is used at the end of a method signature. This is called **declaring** an exception.

The `throws` keyword

The throws keyword

- The `throws` keyword indicates the type of exception that is likely to be thrown by a method
- The method `readAFile` is likely to throw an exception as the file might not be present in the path specified

```
public void readAFile(String filepath) throws FileNotFoundException
{
    // code to read a file
}
```

Handling User-Defined Exceptions

User-defined exceptions can be explicitly thrown from a method. This is done using the `throw` keyword.

The `throw` keyword

The throw Keyword

The throw keyword throws an object of type `Exception` which is handled by an exception handler written in the calling method.

```
if (age < 13) {
    throw new InvalidAgeException("Cannot
register on the platform, not a valid
age");
}
```

Handling User-Defined Exceptions in a Program

When handling user-defined exceptions, the `throws` and `throw` method must be used appropriately as shown below

```
public void verifyAge(int age) throws InvalidAgeException {  
    if (age < 13) {  
        throw new InvalidAgeException("Cannot register on  
the platform, not a valid age");  
    }  
}
```

Age Verification

Most of the social media platforms require users to be at least 13 years of age to use their services.

Write a program to verify the age of the users when registering on a social media platform.

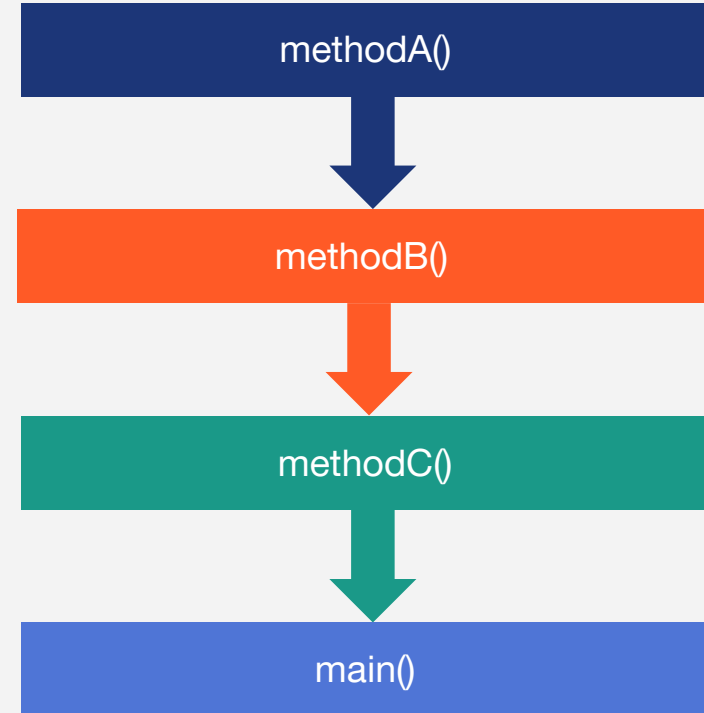
Raise an alert if the age of the user is less than 13 years.



Exception Propagation

- In Java, all methods are populated in the memory in the order in which they are called by the main method
- The exceptions that occur during execution are propagated down the stack of methods until an appropriate exception handler is encountered
- The exception handler is usually a try and catch block

Call Stack



Key Takeaway

- User-defined exceptions
- The throws and throw keywords
- Implementing user-defined exceptions in Java
- Exception propagation





Thank you!