

# Rajalakshmi Engineering College

Name: Abishek D  
Email: 240701013@rajalakshmi.edu.in  
Roll no: 240701013  
Phone: 8825816124  
Branch: REC  
Department: CSE - Section 10  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### REC\_2028\_OOPS using Java\_Week 8\_MCQ

Attempt : 1  
Total Mark : 15  
Marks Obtained : 15

#### **Section 1 : MCQ**

- What will be the output for the following code?

```
class InvalidUsernameException extends Exception {  
    public InvalidUsernameException(String message) {  
        super(message);  
    }  
  
    class Test {  
        public static void main(String[] args) {  
            try {  
                String username = "abc";  
                if (username.length() < 5) {  
                    throw new InvalidUsernameException("Username must be at  
least 5 characters long");  
                }  
            }  
        }  
    }  
}
```

```
        } catch (InvalidUsernameException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

**Answer**

Username must be at least 5 characters long

**Status :** Correct

**Marks :** 1/1

2. What will be the output for the following code?

```
class NegativeBalanceException extends Exception {
    public NegativeBalanceException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            double balance = -500;
            if (balance < 0) {
                throw new NegativeBalanceException("Balance cannot be
negative");
            }
        } catch (NegativeBalanceException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

**Answer**

Error: Balance cannot be negative

**Status :** Correct

**Marks :** 1/1

3. What will be the output for the following code?

```
import java.io.*;

class UnderageException extends Exception {
    public UnderageException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            int age = 17;
            if (age < 18) {
                throw new UnderageException("Underage, cannot proceed");
            }
        } catch (UnderageException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

**Answer**

Underage, cannot proceed

**Status :** Correct

**Marks :** 1/1

4. What will be the output for the following code?

```
import java.io.*;

class NegativeAgeException extends Exception {
    public NegativeAgeException(String message) {
        super(message);
    }
}

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

```
try {
    int age = -5;
    if (age < 0) {
        throw new NegativeAgeException("Age cannot be negative");
    }
} catch (NegativeAgeException e) {
    System.out.println(e.getMessage());
}
```

**Answer**

Age cannot be negative

Status : Correct

Marks : 1/1

5. What will be the output of the following code?

```
class MyException extends Exception {
    public MyException() {
        super("Default Exception Message");
    }
}
```

```
class Test {
    public static void main(String[] args) {
        try {
            throw new MyException();
        } catch (MyException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

**Answer**

Default Exception Message

Status : Correct

Marks : 1/1

6. What is the purpose of a custom exception in Java?

**Answer**

To create user-defined exceptions for specific scenarios

**Status : Correct**

**Marks : 1/1**

7. What will be the output for the following code?

```
import java.io.*;

class TemperatureTooHighException extends Exception {
    public TemperatureTooHighException(String message) {
        super(message);
    }
}

class Test {
    public static void main(String[] args) {
        try {
            int temperature = 110;
            if (temperature > 100) {
                throw new TemperatureTooHighException("Temperature too
high");
            }
        } catch (TemperatureTooHighException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

**Answer**

Temperature too high

**Status : Correct**

**Marks : 1/1**

8. what is the output of the following code?

```
240701013 class MyException extends Exception {  
240701013     public MyException(String message) {  
240701013         super(message);  
240701013     }  
240701013 }  
240701013  
240701013 class Test {  
240701013     public static void main(String[] args) {  
240701013         try {  
240701013             throw new MyException("Error occurred");  
240701013         } catch (MyException e) {  
240701013             System.out.println(e);  
240701013         }  
240701013     }  
240701013 }
```

**Answer**

MyException: Error occurred

**Status : Correct**

**Marks : 1/1**

9. Which of the following is true about custom exceptions?

**Answer**

Custom exceptions must extend either Exception or RuntimeException

**Status : Correct**

**Marks : 1/1**

10. What will be the output for the following code?

```
240701013 class InvalidVotingAgeException extends Exception {  
240701013     public InvalidVotingAgeException(String message) {  
240701013         super(message);  
240701013     }  
240701013 }
```

```
240701013 class Test {  
240701013     public static void main(String[] args) {
```

```
try {  
    int age = 15;  
    if (age < 18) {  
        throw new InvalidVotingAgeException("You are not eligible to  
vote");  
    }  
    System.out.println("Eligible to vote");  
} catch (InvalidVotingAgeException e) {  
    System.out.println(e.getMessage());  
}
```

**Answer**

You are not eligible to vote

**Status : Correct**

**Marks : 1/1**

11. What will happen if a checked custom exception is thrown inside a method without being caught or declared?

**Answer**

Compilation Error

**Status : Correct**

**Marks : 1/1**

12. Which keyword is used to explicitly throw a custom exception?

**Answer**

throw

**Status : Correct**

**Marks : 1/1**

13. What will be the output for the following code?

```
import java.io.*;
```

```
class OutOfStockException extends Exception {
```

```
public OutOfStockException(String message) {  
    super(message);  
}  
  
}  
  
class Test {  
    public static void main(String[] args) {  
        try {  
            int stock = 0;  
            if (stock == 0) {  
                throw new OutOfStockException("Item is out of stock");  
            }  
        } catch (OutOfStockException e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

**Answer**

Item is out of stock

**Status : Correct**

**Marks : 1/1**

14. How do you create an unchecked custom exception?

**Answer**

By extending RuntimeException

**Status : Correct**

**Marks : 1/1**

15. what is the output of the following code?

```
class MyException extends Exception {  
    public MyException(String message) {  
        super(message);  
    }  
}
```

```
class Test {  
    static void check() throws MyException {  
        throw new MyException("Custom Exception Occurred");  
    }  
  
    public static void main(String[] args) {  
        try {  
            check();  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

**Answer**

Custom Exception Occurred

**Status :** Correct

**Marks :** 1/1