MANIPAL ACADEMY of HIGHER EDUCATION (Deemed to be University under Section 3 of the UGC Act, 1956)

Question Paper - Report

10-Oct-2024 14:34:59
ADITYA PANKAJ MATHPAL .

Logout

Back

Question Paper

MANIPAL ACADEMY OF HIGHER EDUCATION

B.Tech III Semester Sessional Examination September 2024

OBJECT ORIENTED PROGRAMMING [CSE 2124]

Marks: 30 **Duration: 90 mins.**

MCQ

Answer all the questions. Section Duration: 20 mins

1) Find the output of the following code snippet. (0.5)

```
class access
{ static int x;
void increment()
{ x++; } }
class static_use
{
public static void main(String args[])
{
access obj1 = new access();
access obj2 = new access();
obj1.x = 0;
obj1.increment();
obj2.increment();
System.out.println(obj1.x + " " + obj2.x);
}
}
```

1 2 1 1 2 2 Compilation Error

2) Identify the output of the following code snippet (0.5)

```
class output
{
public static void main(String args[])
{
String c = "Hello i love java";
int start = 2;
int end = 9;
char s[]=new char[end-start];
c.getChars(start,end,s,0);
System.out.println(s);
}
}
```

Hello, i love java | i love ja | lo i lo | llo i l

3) Determine the output for the following. (0.5)

```
class Test
{
static int a;
static
{
a = 4;
System.out.println ("inside static block\\n");
System.out.println ("a = " + a);
}
Test()
{
System.out.println ("\\ninside constructor\\n");
a = 10;
}
public static void func()
{
a = a + 1;
System.out.println ("a = " + a);
}
public static void main(String[] args)
{
Test obj = new Test();
obj.func();
}
}
```

inside static block a=4 inside constructor a=11	inside static block a=4 inside constructor a=5	inside static block a=10 inside constructor a=11	Run time Error

4) Find the output for the following (0.5)

```
class Base {

public final void display() {

System.out.println("Display from Base");

}

}

class Derived extends Base {

public void display() {

System.out.println("Display from Derived");

}

}

public class Test {

public static void main(String[] args) {

Base b = new Derived();

b.display();

}

}
```

Display from Derived   Display from Base   Compilation Error   Runtime Error

5) Predict the output of the following program. (0.5)

```
class A

{

public void print() { System.out.print("A"); }

}

class B extends A

{

public void print() { System.out.print("B"); }

}

class Main

{

public static void Doprint( A O)

{ O.print(); }

public static void main(String args[])

{

A x = new A();

B y = new B();

B z= new B();

Doprint(x);
Doprint(y);
Doprint(z);
}
}
```

BBA BAB ABB AAA

6) The ____________ method allows run-time polymorphism in Java. (0.5)

Overloaded Overridden Same class Different parameter types but with the same name

7) When is the object created with new keyword? (0.5)

At run time At compile time Depends on the code When the program is loaded

8) Identify the true statements in Java. (0.5)

i) Enables the creation of cross-platform programs

ii) Can resolve type information at run time

iii) Does not support memory management

iv) Execution is confined to JVM

All are true i, ii. iii i, ii, iv ii, iii, iv

9) Identify the correct statement(s) about the finalize() method. (0.5)

Used for specific actions that will occur when an object is just about to be reclaimed by the garbage collector can be added to a class, by simply defining the finalize( ) method used to free non-java resources such as file handle or window character font All the options

10) Determine the output of the following code. (0.5)

```
class Test {
public static void main(String args[]) {
int arr[2];
System.out.print(arr[0]);
System.out.println(arr[1]);
}
}
```

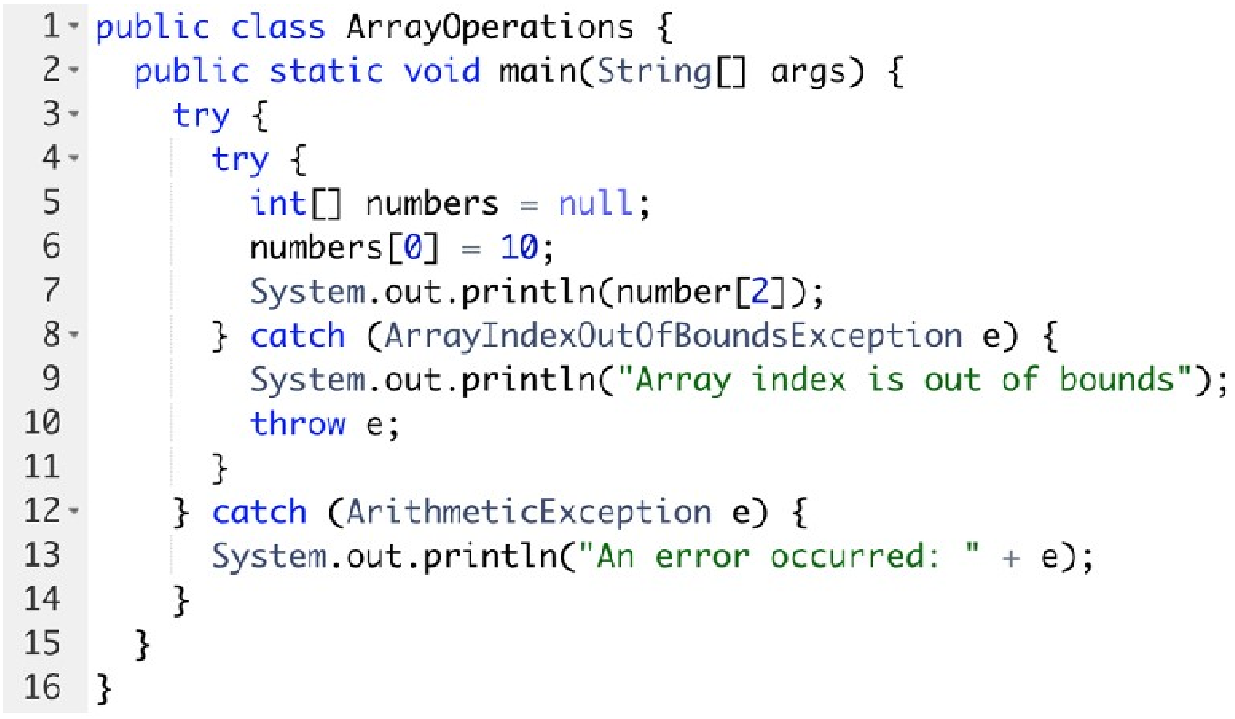
Compile Error garbage value garbage value Exception 0 0

DESCRIPTIVE

Answer all the questions.

1) Outline the advantages of Packages in Java. (2)

2) Consider the below Java program that contains several issues in how exceptions are handled during array operations. Identify the errors (line nos) and rewrite the code. (2)



```java
public class ArrayOperations {
  public static void main(String[] args) {
    try {
      try {
        int[] numbers = null;
        numbers[0] = 10;
        System.out.println(number[2]);
      } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("Array index is out of bounds");
        throw e;
      }
    } catch (ArithmeticException e) {
      System.out.println("An error occurred: " + e);
    }
  }
}
```

3) Demonstrate how a nested interface can be implemented. (2)

4) Write a Java program that reads a student's name and year of graduation. If the graduation year is a leap year(*Note: Use the logic to check the leap year*), a custom exception should be thrown from **CheckLeapYear ()** with an appropriate message that displays the leap year. (3)

5) Develop a Java program to manage the library system, where each book has the attributes: title, author, and ISBN (all strings). The program should allow the librarian to enter the details of up to five books, storing them in an array of Book objects. The program should include two methods one for displaying book details and the other for searching a book by its partial title, and if a match is found, display the corresponding book(s) details. (3)

6) An interface **MyConstants** is defined as follows: (3)

package mypackage;

public interface MyConstants {

int ANSWER = 42;

}

Demonstrate two ways to bring the constant ANSWER into view for use in a class **ExamDemo** (residing outside *mypackage*) that contains the **main()** with appropriate justifications**.**

7) Write a method checkArmstrong(int) to check if a number is Armstrong (*a number that equals the sum of its digits, each raised to a power*) or not. Create a class ComputeDemo to input lower and upper limits and display all Armstrong numbers between these limits using the checkArmstrong(int) method and find the sum of these Armstrong numbers in the main() method. (3)

Note: Consider any n-digit number.

8) Create a Java program to simulate bank operations using exception handling. First, define three user-defined exceptions: the base class **InvalidTransactionException**, and two subclasses, **InsufficientBalanceException** (triggered when the withdrawal amount exceeds the account balance) and **DailyLimitExceededException** (triggered when the withdrawal exceeds a daily limit of 50,000 units). In (3)

the **BankTransaction** class, prompt the user to enter **account_no**, **current_balance**, and **withdrawal_amount**. Use nested **try-catch** blocks where the outer block checks the correctness of the **account_no**, and the inner block checks for both **InsufficientBalanceException** and **DailyLimitExceededException**. Display relevant error messages when exceptions are caught by suitably using the constructor of **Exception** class.

Note: Use the default value "12345678" for the valid account number and set the daily withdrawal limit to Rs. 50,000.

9) A company pays its employees on a weekly basis (Consider company as an abstract class and empName as a private variable). The employees are of three types: i) Salaried employees are paid a fixed weekly salary regardless of the number of hours worked, ii) hourly employees are paid by the hour and receive overtime pay (i.e., 1.5 times their hourly salary rate) for all hours worked in excess of 40 hours, iii) commission employees are paid a percentage of their sales. Design the class hierarchy, include the constructors in each class with appropriate instance variables, make the instance variables private and method to print the employee name and salary. Write a test class to show all the functionalities. Use dynamic method dispatch. (Note: make suitable assumptions for data) (4)