
Mid Term Examination- I, Spring Semester 2024-2025

Program Name: B. Tech. – CSE

Year: II year

Course Code: CS 23.202

Max. Mark: 20

Course Name: Advanced Programming

Weight: 100%

Total no. of Printed Pages: 2

Time: 60 Minutes

Instructions to candidates:

1. Question number 1 is compulsory; each short answer type question is of 1 mark.
 2. Attempt any three questions out of four (Question number 2 to 5); each question is of 5 marks.
-

Q1. Write one (only one and with minimal exemplification) major difference between

- a). static and non-static method in terms of its implementation.
- b). operator overloading and method overloading in terms of java feature.
- c). default and parametric constructor in terms of implementation.
- d). method overloading and method overriding in terms of programmer friendliness.
- e). primitive and user-defined data types in terms of permissible operations.

a). A static method belongs to the class and therefore object creation and calling using dot operator is not required. Non-static members are required to be accessed by object using dot operator.

b). Java doesn't support the feature of operator overloading as compared to method overloading

c). Parametric constructors necessitate the programmer to provide at least one parameter during declaration of the constructor.

d). Method overriding is less programmer friendly as compared to method overloading

e) User defined data types such as Integer and String class provide built-in class methods and provide tremendous functionalities as compared to primitive data types and are therefore more useful and allow more operations.

Q2. Following Java program asks the user's year of birth and current year and calculates his/her age. Analyze the code and answer the following questions.

```
1 public class AgeCalculator{
2     static int processor(int a, int b){
3         int age = a - b;
4         return(age);
5     }
6     static int blackBox(int age){
7         int processed_age = age;
8         /* You may write the code here to,
9            Processed_age = Math.abs(age);
10            process raw data 'age' field */
11         return(processed_age);
12     }
13     public static void main(String[] args) {
14         int yb = Integer.parseInt(args[0]);
15         int cy = Integer.parseInt(args[1]);
16         int age = processor(yb, cy);
17         System.out.print("Value of calculated age is = ");
18         System.out.print(blackBox(age));
19     }
20 }
```

Code is compiled and executed with following lines below

```
javac AgeCalculator.java
java AgeCalculator 1989 2025
```

Q2.1 Suggest how the efficiency of the code can be improved. [2 marks]

Q2.2 Suggest what can be written in **black_box()** method to improve the output in terms of utility, readability and insights [3 marks]

Q2.1 There are following possible modifications.

1. The age variable is not supposed to hold a value greater than 127, therefore 'byte' data type is more efficient for age variable.
- ~~2. The order of variable yb and cy can be swapped to get rid of negative value for age~~
- ~~3. The formula in line no. 3 can be reversed 'int age = b - a'~~

Q2.2 In the blackBox method, age variable could be passed to absolute method 'processed_age = Math.abs(age)'.

This will make the code robust, to handle either order of year-of-birth and current-year.

Q3. Analyze the java code given below and answer the following questions.

3.1 Briefly explain the line no. 12 which contains **this.x = super.x** as if it is not a comment but a valid java code. [1 mark]

3.2 Looking at line no.11 and 12, explain the difference between **super()** and **super.x** functionality, as if it is not a comment. [1 mark]

3.3 Write the exact output of following Java code as printed on console. [3 marks]

```
1 class A{
2     String x = "A";
3     A(){
4         // String x = x.toString();
5         System.out.println("A constructor called, and value of x
        is = " + x);
6     }
7 }
8 class B extends A{
9     String x = "B";
10    B(){
11        // super();
12        // this.x = super.x;
13        int y = x.length();
14        System.out.println("B constructor called, and value of
        x is = " + y);
15    }
16    void show(){
17        System.out.println("It's a method, not a constructor");
18    }
19 }
20 public class question {
21     public static void main(String [] args){
22         B b = new B();
23         b.show();
24     }
25 }
```

Q 3.1 This assign value x of A class to variable x of B class.

Q 3.2
super() calls the constructor of parent (super) class. Super.x is used to access x variable of parent class.

Q 3.3 A constructor called, and value of x is = A
B constructor called, and value of x is = 1
It's a method, not a constructor

Q4. Analyze the Java code given below and briefly explain each line of the code in terms of the concept it demonstrates. Write the exact output of the code as printed on the console.

```

1  class A{
2      A() {
3          Integer a = 13;
4          System.out.print(a+"-"); }}
5  class B extends A{
6      B() { super();
7          System.out.println("B constructor");}}
8  class Const{
9      public static void main(String[] args){
10         A a = new B(); }}

```

Answer Q4. The question is based on inheritance and type casting (We perform upcasting in line no. 10) – Casting super class to subclass type. First we have defined a class A, in which we have defined an object a, which belongs to Integer class and we have assigned it a value 13. Class B inherits class A and its constructor has been called. This calls A() constructor first and then B() constructor. This results in printing

13-B constructor

Q5. Analyze the following java code, populate the simulation table and write the exact output of the code as printed on the console.

```

1  public class Simulation {
2      static int simulacra(int a, int b){
3          for (int i = a ; i < b; i++) {
4              b = b + 10; i = i + 20;
5              System.out.print("a = " + a + "; b = "+ b);
6              System.out.println("; i = " + i);
7          }
8          return(b);
9      }
10     public static void main(String[] args) {
11         int l0 = 50; int o1 = 20; int b = simulacra(o1, l0);
12     }
13 }

```

Answer:

	l0	o1	b	i	a	b
11	50	20				
3	50	20		20	20	50
4	50	20		40	20	60
3	50	20		41	20	60
4	50	20		61	20	70
3	50	20		62	20	70
4	50	20		82	20	80
11	50	20	80	82	20	80

Output of the code

```

a = 20; b = 60; i = 40
a = 20; b = 70; i = 61
a = 20; b = 80; i = 82

```