

**ICSE PRE SEMESTER-1 RE EXAMINATION  
COMPUTER APPLICATIONS**

---

**MAXIMUM MARKS: 50**

**TIME ALLOWED: 1:00 HOURS (Inclusive of reading time)**

**ALL QUESTIONS ARE COMPULSORY**

The marks intended for questions are given in the brackets [ ]

---

*Select correct option for each of the following questions.*

---

**SECTION-A (30 MARKS)**

**QUESTION 1**

**[5\*1=5]**

- a) A class is :
- 1) Memory allocated to an object.
  - 2) Variables allocated to an object.
  - 3) Methods allocated to an object.
  - 4) A blueprint that defines the variables and the methods common to all objects of a certain kind.
- b) \_\_\_\_\_ are either Java reserved words or titles given to variables, classes and methods.
- 1) Keywords            2) Operators            3) Punctuators            4) Identifiers
- c) The Java programming language has a term for its building blocks, or basic elements:
- 1) Literals            2) Operators            3) Tokens            4) Methods
- d) In Java, strings are handled by a special class called:
- 1) Character            2) char            3) String            4) str
- e) The modifier \_\_\_\_\_ states that the method is accessible to other classes.
- 1) void            2) public            3) static            4) main

**QUESTION 2**

**[5\*1=5]**

- a) \_\_\_\_\_ variables are the variables that contain the state of an object.
- 1) Instance            2) Object            3) Class            4) Hybrid
- b) In Java, if your class does not extend any other class, it still extends:
- 1) Java.lang.Object    2) Java.lang.nodes    3) Java.lang.Classes    4) Java.lang.instances
- c) \_\_\_\_\_ inheritance does not exist in Java.
- 1) Multiple            2) Hybrid            3) Multilevel            4) Single
- d) Primitives that are fields in a class are automatically initialized to:
- 1) Zero            2) One            3) Two            4) Three
- e) Similar to primitives, the object references are initialized to:
- 1) Zero            2) Nil            3) Null            4) None of these

**QUESTION 3**

**[5\*1=5]**

- a) There are two variables of methods defined in a class. They are instance methods and \_\_\_\_.
- 1) Program methods            3) Skipper methods
  - 2) Runner methods            4) Class methods
- b) A private method is accessible only:
- 1) To classes in the package            3) To any class anywhere
  - 2) Within the class            4) None of these
- c) What does ++ operator do in Java?
- 1) Increments value by 1            3) Decrements value by 1
  - 2) Inverts the value of a Boolean            4) Inverts the sign of an expression
- d) \_\_\_\_\_ is the first line of the function definition that describes the return type of the output, the number and types of arguments to be passed and any throws, extends or implements clauses.
- 1) Function definition            3) Function signature
  - 2) Function prototype            4) Function invocation
- e) In a function, \_\_\_\_\_ contains a collection of statements that define what the method does.
- 1) Return statement            3) Parameters list
  - 2) Method body            4) All of the above

**QUESTION 4**

**[5\*1=5]**

**Study the code below and answer the following questions:**

```
class Hello
{
    int abc, def;
    Hello()
    {
```

```

        abc = 1;
        def = 2;
        System.out.println("Default Constructor built");
    }
    Hello(int a, int b)
    {
        abc = a;
        def = b;
        System.out.println("Another Constructor built");
    }
}
public class Starter
{
    public static void main(String args[ ])
    {
        Hello ob = new Hello( );
        System.out.println("abc = "+abc);
        System.out.println("def = "+def);
    }
}

```

- a) What is the first line of the output for the given code snippet?
- 1) Object Created
  - 2) Default Constructor built
  - 3) Another Constructor built
  - 4) None of the above
- b) What type of constructor is invoked first by the code?
- 1) Non Parameterized Constructor
  - 2) Default Constructor
  - 3) Parameterized Constructor
  - 4) None
- c) If the line **Hello ob = new Hello( )**, would be replaced by **Hello ob = Hello(2, 3)**, then what constructor would be invoked?
- 1) Default Constructor
  - 2) Parameterized Constructor
  - 3) Both
  - 4) None
- d) What will be the value of **abc**?
- 1) 1
  - 2) 2
  - 3) 5
  - 4) -1
- e) What will be the value of **def**?
- 1) 2
  - 2) 1
  - 3) -1
  - 4) 3

#### QUESTION 5

[5\*1=5]

Choose the odd one out:

- |                  |                 |                |                |
|------------------|-----------------|----------------|----------------|
| a) 1) Class      | 2) Array        | 3) int         | 4) Interface   |
| b) 1) &&         | 2)              | 3) !           | 4) ==          |
| c) 1) +=         | 2) -=           | 3) ==          | 4) *=          |
| d) 1) parseInt() | 2) parseFloat() | 3) parseLong() | 4) parseChar() |
| e) 1) \t         | 2) \n           | 3) \a          | 4) \'          |

#### QUESTION 6

[5\*1=5]

State true/false:

- a) Constructors are only used to initialize instance variables.
- 1) True
  - 2) False
- b) **this** keyword is used to allocate memory to the object.
- 1) True
  - 2) False
- c) Data type **long** occupies **8 bytes** in memory.
- 1) True
  - 2) False
- d) Output of **a++ + ++a + a** will be **11** if, **a=2**
- 1) True
  - 2) False
- e) Ternary Operator uses more than 2 operands.
- 1) True
  - 2) False

### SECTION-B(20 MARKS)

#### QUESTION 7

[6\*1=6]

Given below is a class with the following specifications:

**Class name:** Fibonacci

**Member Methods:**

int cal(int n) : to find the sum of the Fibonacci series upto n terms

void printSum(int s) : to print the sum of the series

Fill in the blanks of the given program with appropriate Java statements:

```
class ____(a)____
{
    int cal(int n)
    {
        int x=0, y=1, c, sum =0;
        int i;
        for(i = 1 ; ____(b)____ ; i++)
        {
            c = ____(c)____;
            a = b;
            b = c;
            sum = sum + ____(d)____;
        }
        return ____(e)____;
    }
    void printSum(int sum)
    {
        System.out.println("The sum of the series is: " + ____(f)____);
    }
}
```

- |                 |              |              |
|-----------------|--------------|--------------|
| a) 1) Fibonacci | 2) FIBONACCI | 3) Fibonacci |
| b) 1) i < n     | 2) i <= n    | 3) i > n     |
| c) 1) a         | 2) b         | 3) a + b     |
| d) 1) c         | 2) b         | 3) a         |
| e) 1) a + b     | 2) c         | 3) sum       |
| f) 1) a + b     | 2) c         | 3) sum       |

**QUESTION 8**

[6\*1=6]

Given below is a class with the following specifications:

- |                         |  |
|-------------------------|--|
| class name              | : Result   |
| member variables:       |  |
| int phy                 | : to store the marks of physics  |
| int che                 | : to store the marks of chemistry  |
| int bio                 | : to store the marks of biology  |
| member methods:         |  |
| result( )               | : default constructor to initialize member variables as 0                                  |
| result(int, int, int)   | : parameterized constructor to initialize values to instance variables                     |
| double calculate( )     | : find the average of all the marks.   |
| void displayRes(double) | : to display whether the student is pass or fail according to the details mentioned below: |

AVERAGE	RESULT
avg < 40	fail
avg >= 40 and less than 60	Pass with II division
avg >= 60 and less than 75	Pass with I division
avg >= 75	Pass with Distinction

Fill in the blanks of the given program with appropriate Java statements:

```
class ____(a)____
{
    int phy, chem., bio;
    Result()
    {
        phy = 0;
        che = 0;
        bio = 0;
    }
    Result(int a, int b, int c)
    {
        phy = a;
        che = b;
        bio = c;
    }
    double calculate( )
    {
        double avg = ____(b)____;
        return ____(c)____;
    }
}
```

```

void displayRes(double avg)
{
    if(__(d)__)
        System.out.println("fail");
    else if(avg >= 40 __(e)__ avg<60)
        System.out.println("pass with II division");
    else if( avg >= 60 && avg < 75 )
        System.out.println("pass with I division");
    else if( avg >= 75)
        System.out.println("__(f)");
}

```

- |                             |                    |           |
|-----------------------------|--------------------|-----------|
| a) 1) result                | 2) RESULT          | 3) Result |
| b) 1) (a+b+c)/2             | 2) (a+b+c)/3       | 3) a+b+c  |
| c) 1) avg                   | 2) bio             | 3) phy    |
| d) 1) avg>40                | 2) avg<=40         | 3) avg<40 |
| e) 1)                       | 2) &&              | 3) !      |
| f) 1) Pass with Distinction | 2) Pass with Grace | 4) Pass   |

### QUESTION 9

[4\*1=4]

**Design a class to find the factorial of a number.**

Factorial of a whole number 'n' is defined as the product of that number with every whole number till 1.

**For e.g. factorial of 4 = 4\*3\*2\*1 = 24**

```

import java.io.*;
class Fact
{
    void __(a)___
    {
        int i, f = 1;
        for( i = 1; i<=n; i++)
            f = __(b)___ ;
        System.out.println("factorial of "+n+" is: "+ __(c)___);
    }
    public static void main(int n)
    {
        __(d)___ ob = new Fact();
        ob.factorial(n);
    }
}

```

- |                   |                     |                |                 |
|-------------------|---------------------|----------------|-----------------|
| a) 1) factorial() | 2) factorial(int n) | 3) factor(int) | 4) factorial(n) |
| b) 1) i*f         | 2) i*i              | 3) f*f         | 4) i            |
| c) 1) f           | 2) i                | 3) i*i         | 4) f*f          |
| d) 1) Factorial   | 2) Fact             | 3) fact        | 4) FACT         |

### QUESTION 10

[4\*1=4]

#### CASE STUDY

**Read the following paragraph and answer the following questions:**

The class is a blueprint to create objects in a program. Every object typically contains attributes as well as methods. The attributes are also known as data members. These data members keep track of the state of an object. Also contained within the class definition are methods which can be invoked in order to change the attribute or data values. Methods also help to pass messages to objects. Suppose there is an object. These objects can be passed messages via methods and there are methods that react to this message passed. It is like a human being. You have part of your body and their various things you can do with your parts. The things or actions you can do are the methods of an object. Abstract classes are classes that have no implementation. These classes have vague designs which are implemented when these classes are inherited and extended.

- |  |                 |              |                        |               |
|--|-----------------|--------------|------------------------|---------------|
| a) What is the name of classes that have no implementation?                      | 1) Vague        | 2) Diffident | 3) Recurrent           | 4) Abstract   |
| b) When a message is passed to an object, it often react with a :                | 1) Class        | 2) Object    | 3) Method              | 4) Stunt      |
| c) What aspect of class, or an object created points to the state of the object? | 1) Attributes   | 2) Methods   | 3) Abstract attributes | 4) Definition |
| d) What is the other name given to the attributes of a class?                    | 1) Data members | 2) Methods   | 3) Abstract class      | 4) None       |