Chhatrapati Shahu Ji Maharaj
University, Kanpur

Answer Script Details
Barcode 11948414

Roll No. 23071002365
Total Mark 50/75.00

Exam BCA-V_ODD_EXAM_NOV_2025
Subject BCA5002 - JAVA PROGRAMMING AND DYNAMIC WEB

Question wise Mark Summary

Q.No	Mark	Q.No	Mark	Q.No	Mark	Q.No	Mark
1A	3.5/5						
1B	3/5						
1C	3/5						
1D	2.5/5						
1E	3.5/5						
1F	3/5						
1G	3.5/5						
1H	3/5						
1I	3/5						
2	11/15						
3	0/15						
4	0/15						
5	0/15						
6	0/15						
7	0/15						
8	0/15						
9	11/15						

Chhatrapati Shahu Ji Maharaj University
Kanpur, Uttar Pradesh

PART-I

Date of Exam : 17-11-25 Shift Evening Room No. 118
Paper Code BCA 5002 Subject Java Programming and dynamic webpage Design Year/Sem. V
Name of Candidate Khyati Trivedi
Roll No. 2307100 2365

Signature of Candidate | Signature of Invigilator | COE Facsimile

PART-II

MARKS OBTAINED

Q.	1	2	3	4	5	6	7	8	9	10
(a)										
(b)										
(c)										
(d)										
(e)										
(f)										
(g)										
(h)										
(i)										
(j)										
Total										

Total Marks in Figures — Max. Marks

Total Marks in Words

BCA5002
Paper Code

Signature of Evaluator

PART-III

Course Bachelor of Computer Application

Session 2025-2026 Year/Semester V

Subject Java Programming and dynamic webpage Design

Paper Code: BCA5002

Exam Date: 17 11 2025

Name of Candidate: KHYATI TRIVEDI

Father's Name: SHAILENDRA KUMAR TRIVEDI

महाविद्यालय का कोड College Code: KN162

परीक्षा केन्द्र का कोड Exam Centre Code: KN162

परीक्षा का प्रकार Type of Exam: संस्थागत Regular ● / भूतपूर्व छात्र Ex Student ○ / व्यक्तिगत Private ○ / बैक पेपर परीक्षा Back paper Exam ○

ANSWER BOOKLET NO. 11948414

BCA5002
Paper Code

PART-IV

नामांकन संख्या Enrollment Number: CSJMA23000129517

परीक्षार्थी अनुक्रमांक संख्या Candidate's Roll Number: 23071002365

पेपर कोड Paper Code: 5002

Signature of Candidate

Signature of Invigilator

C S Facsimile

COE Facsimile

नोट : 1. परीक्षार्थी को निर्देशित किया जाता है कि आवरण पत्र के पृष्ठ भाग पर अंकित सभी निर्देशों को सावधानी पूर्वक पढ़ें। 2. बॉक्स में भरी जाने वाली प्रविष्टियाँ बायीं तरफ से शुरू की जायें। 3. गोलों को काले या नीले बॉलपेन से भरा जायें।

INSTRUCTIONS TO THE CANDIDATE FOR FILLING PART-I

1. Read the instructions carefully given on the answer script and admit card.
2. Write Date of Exam, Shift, Paper Code & Name of Subject Correctly.
3. Write Name & Roll No. Correctly.
4. Write Semester & Branch Correctly.

INSTRUCTIONS TO THE CANDIDATE FOR FILLING PART-III

1. Use blue or black ball point pen for writing alphabets & numerals in ☐ Boxes.
2. Carefully study the example before you start marking.
3. As shown in the example below blacken the circles completely.

Correct method	Incorrect method
① ● ③ ④	✓ ✗ ● ◎

4. Make no Stray marks on this sheet.
5. **DO NOT WRITE OR MARK ON THE BAR CODE.**

परीक्षार्थी के लिए निर्देश

1. प्रवेश पत्र एवं उत्तर पुस्तिका पर दिये गये निर्देशों को ध्यान से पढ़ें।
2. कवर पृष्ठ के दूसरी तरफ कुछ न लिखें।
3. उत्तर पुस्तिका के पृष्ठों पर दोनों तरफ लिखें।
4. प्रश्न पत्र पर अपने अनुक्रमांक के अतिरिक्त कुछ न लिखें।
5. प्रश्न पत्र कोड एवं प्रश्न पत्र कोड सावधानी पूर्वक लिखें।
6. अपनी स्थिति स्पष्ट लिखें।
7. उत्तर पुस्तिका के पृष्ठों की संख्या देखें। अगर उत्तर पुस्तिका में पृष्ठ (1–24) से कम है या फटे हुए है, तो परीक्षा शुरू होने के पूर्व दूसरी उत्तर पुस्तिका ले लें।
8. प्रश्नपत्र को देख, यदि प्रश्नपत्र के विषय कोड, विषय का नाम तथा प्रश्न में कोई त्रुटि है तो उसके परीक्षा शुरू होने के 30 मिनट के अन्दर कक्ष निरीक्षक को तत्काल सूचित करें, उसके बाद विश्वविद्यालय द्वारा कोई कार्यवाही नहीं की जायेगी।
9. प्रश्नों के उत्तर लिखने के लिये पेंसिल का प्रयोग न करें।
10. B कोपी या अतिरिक्त ग्राफ नहीं दिया जायेगा।

IN ORDER TO AVOID UFM (UNFAIR MEANS) :

1. The Roll No. and Answer Book no. found elsewhere or any other symbol found in the answer book will be treated as unfair means.
2. Any tempering of Bar Code and Booklet no shall be treated as Unfair Means.
3. Do Not bring the materials like slip of paper/mobile/digital diaries/ study material/ revision notes in examination hall. Possession of the mobiles/ digital diaries/ electronic watch and any other electronic gadget except memory less scientific calculator shall be considered as UFM case.
4. Do not keep or paste currency note in answer script it shall be consider as UFM.

INSTRUCTIONS TO THE CANDIDATE

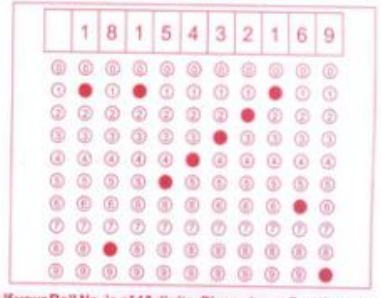
1. Read the instructions carefully given on the Question Paper Admit Card & Answer Script.
2. Do not write anything on back side of the cover page.
3. Write on both sides of pages of answer book.
4. Do not write anything on question paper except Roll Number.
5. Write Paper Code & Question Paper Id carefully.
6. CHECK the number of pages (1-32) or any other kind of damage in your answer script, if found than change the answer script immediately before the commencement of examination.
7. CHECK the Question Paper for any kind of discrepancy e.g. Subject Code, Subject Name and Question of the Question Paper during first THIRTY MINUTES of the commencement of the exam, so that it can be corrected in TIME. After that no corrections shall be entertained by the university.
8. Do not use pencil for answering the question.
9. Write status correctly e.g. those appearing in carry over papers should fill in status as Carry Over. Those appearing as Ex Students should fill in status as ex.
10. No supplementary answer book & graph paper will be provided.

अनुचित साधन से बचने हेतु:

1. उत्तर पुस्तिका के निर्दिष्ट स्थान को छोड़कर अनुक्रमांक एवं उत्तरपुस्तिका का क्रमांक कहीं और न लिखे तथा कोई भी चिन्ह न बनाये क्योंकि यह अनुचित साधन प्रयोग की परिधि में आता है।
2. उत्तर पुस्तिका के बारकोड अथवा उत्तर पुस्तिका संख्या पर छेड़ करने पर अनुचित साधन प्रयोग माना जायेगा।
3. परीक्षा कक्ष में निम्न वस्तुएं साथ न लायें, जैसे लिखे हुए कागज के टुकड़े, मोबाईल, डिजिटल डायरी, कोपी, पुस्तक यह सभी वस्तुएं जो अनुचित साधन के अन्तर्गत आती है। केवल संबंधित प्रश्नपत्र में ही मेमोरी लेस साइंटफिक केल्कुलेटर ले जाने की अनुमन्यता होगी।
4. उत्तर पुस्तिकाओं में रुपये न रखें न ही उत्तर पुस्तिका में चिपकायें। ऐसा करना अनुचित साधन प्रयोग की परिधि में आता है।

INSTRUCTIONS TO THE CANDIDATE FOR FILLING PART-IV

1. Use blue or black ball point pen for writing alphabets & numberals in ☐ Boxes.
2. Use blue or black ball point pen for filling the circles.



Note - If your Roll No. is of 10 digits. Please leave first three columns

Paper Code

01

Section-A

Answer-a

C++	Java
1. C++ supports both procedural & object oriented programming paradigm.	Java support the object-oriented programming paradigm & treats everything in class-object relation.
2. It is less safe because it support direct memory access with pointer	No pointers are there in Java therefore it is more secure.
3. C++ supports multiple inheritance	Java does not support multiple inheritance. We have to use Interface in order to extend more than one class.
4. C++ has copy constructor Inbuilt	In Java we don't have copy constructor but we can achieve it using Object.clone() method or manual copy constructor.

5.	C++ does not have a garbage collector.	Java has automatic garbage collection method that saves a lot of memory & make it more efficient for programming.
6.	C++ is less secure	Java is more secure.

Answer - b).

- Polymorphism is a object oriented pillar.
- "Whenever an object / method has more than one form then it is said to be having polymorphism."
- In polymorphism there can be more than one implementation of a single named method / constructor or object.
- In Java, polymorphism is divided into two subparts.

Polymorphism

↓ Compile-time Polymorphism

↓ Run-time Polymorphism

- Compile time polymorphism.
- It is also called early binding.
- When the multiple form mechanism & its binding happens at compile time it is compile time polymorphism.
- Examples :- Method overloading, Constructor overloading.

eg.

```
void add (int a, int b)
{
}

double add (double a, double b)
{
}
```

same name function with different function signatures.

- Runtime Polymorphism.
- Late Binding
- Whenever the multiple form / Implementation & its binding occurs at runtime then it is runtime polymorphism
- Example :- function overriding.

eg.

```
class A
{
    void run() { }
}
class B extends A
{
    void run() { }
}
```

```
void main (String args[])
{
    B obj = new B();
    obj.run(); // derived class run() is called at runtime
}
```

Answer-1 Interfaces

- Interfaces are a blueprint for the classes with the abstract implementations & declarations of the methods / behaviours or properties.
- Interfaces help in 100% abstraction in java programming because —

1). All methods in Interfaces are abstract methods.

2). All variables in interface are final and static.

- Interfaces are used to perform multiple inheritance because it cannot be performed normally in Java because it cannot extend more than one class.
- We can extend the Interfaces using Implements keyword.
- An interface doesnot have constructors as it cannot be instantiated.
- If an Intufac Implements other Interface it is also done using ~~Implements~~ & extends keyword.

- we can make interface with
syn:— interface interface_keyword
{
}

- and it can be implemented with Syntax—
class A implements interface_name
{ // override functions / properties here.
}

eg — runnable interface
callable interface.

Answer - d).

```
Import java.util.Scanner.*;

class Conversion
{
 public static void main (String args[])
  { Scanner Sc = new Scanner
   Scanner sc = System.in("Input a String");
   Scanner sc = new Scanner ("System.in");
   System.out.println ("Input a string") // for input.
   String str=new String(); // Valid input Syntax.
   str = sc.nextLine ();
   String converted="";
   for (int i=0; i<str.length(); i++).
   {
    char ch = str.charAt(i);
    if (ch.isLowerCase()) Character.isLower(asc(ch))
    {
```

```
        converted += character.toUpperCase(ch);
    } // for converting LowerCase Stdy to UpperCase
    else
    {
        converted += character.toLowerCase(ch);
    } // for converting UpperCase to LowerCase.
} // loop ended.
System.out.println("Received:" + str);
System.out.println("Converted:" + converted);
} // main ended
} // class ended.
```

Input: aPPle aDAy.
Output: Appl... dAy.

Answer - e

Exception handling

- Exception handling in java is a mechanism that can identify and handle the runtime exceptions and help in continuous flow of the program without being interrupted.
- There are 5 keywords that are used to handle the exceptions -

1. **try**
   This block is the place where the exception prone content is kept & if any exception is found it is

directly thrown & is handled later.
It cannot be placed alone.

try
{ //code
}

2. catch:
- There can be multiple catch blocks for single try block.
- They are the main block that handle & error & resolve / handle them without termination.

catch (Exception e)
{ //code.
}

3. finally:
- It is a block that is compulsorily executed even if error was found or not.
- We keep our important code here that cannot be skipped.
- finally {
}

4. throw:
- The keyword that creates a Exception object inside the try block & throws or passes that exception to further block if error is found.

5 (throws:)
- This is used with method signature.
- It is just a warning that the method may / may not have error.

ex. class A
{
public static void hello() throws IOException
{
// ...
}
public static void main()
{
A obj;
try { obj.hello(); }
catch (Exception e)
{ Sop(e); }
}

Example of Exception handling

class Mathematics
{
public static void main (String args[])
{
int a=5;
int b=0;
try {
int c = a / b;
}
catch (ArithmaticException e)
{
} System.out.println ("exception found");

finally {
System.out.println("Execution completed");
}
}
}
// Exception found

Answer - 1.

(Multithreading)

- Multithreading in Java is a process in which more than one threads (parts of a process) complete their execution at the same time.
- These threads share same address space but have different execution path.
- We can perform Multithreading with 2 ways:

1. Extending Thread class.

Here the thread class is extended & the run() method is overridden. From base class the start() methods invokes multithreading.

```
class A extend Thread
{
public run()
{
for(int i=0; i<5; i++)
{ System.out.println(i); } }
```

```
class B
{
public static void main (String args[]).
{
   A obj = new A(); //here a thread is
   obj.start();      created.
}
}
```

(2). By implementing runnable interface.

- Here runnable interface is implemented & run() method is overriden & start() from base class invoke thread.
- Here an object to the class is passed to thread.

```
class A implements runnable.
{
  public run()
  { System.out.println("run method");
  }
}
class B
{ public static void main( String args[])
  {
    A obj = new A();
    Thread t = new Thread(obj);
    t.start(); // thread creat.
  }
}
```

- Advantages

(1) A threads improves productivity
(2) Threads helps in sharing resources.
(3) Faster execution.

Answer-g (AWT)

- AWT stands for Abstract Window Toolkit.
- It is a package in Java that offers various components that are used in development of graphics.
- AWT are platform dependent which means that their look and feel vary from OS to OS and device to device.
- There are 2 libraries that are used for awt graphics.

(1) java.awt;
(2) java.awt.event;

There are various awt components that give a interactive UI to java applications.

(1) Containers.
There are the window object that store another small components.

- they do not have title, menu etc.

(2). Frames.
- Frames are the sub-containers that have title, sidebars, menus etc.
- The default layout is Border layout.

(3) Button.
- Button components add buttons to UI
- Used for handling & performing events.

(4). List
- This adds list items on the UI.

eg: of a list

```
import java.awt.Frame.*;
import java.awt.*;
class AWTpractice class frame extends Frame
{
public static print (Graphics g)
{
    g.drawString("Hello
public void print ()
{
    Frame f = new Frame ();
    f.setBounds (200,300);
    f.setVisible ();
    f.setLayout (Null);
}
}
```

Answer-1 (Session tracking).

- Session tracking is a mechanism used in java to store user logs so that the same user can login in a stipulated time without a database.
- Since HTTP is a stateless protocol which means that the user logs are not stored so we can store these logs for efficient storage & utilization in the sessions.
- Servlets in java have a type called HTTP servlets which give the advantage of sessions.
- Sessions are a small data pouch that stores a user's log for a stipulated period of time & within that time no regular verification overhead is performed.
- ~~syntax~~ : we can use session in java, using the HTTPServlet. which provide the HTTP sessions methods.

Syntax :

```
HttpServletSession = n
HttpSession s = new HttpSession();
```

```
HttpSession st = new HttpSession();
st.setAttribute("name", "John");
```

- Advantages.

1) Advantages of using sessions is that without any Database we can store user logs which is memory efficient.

2. Sessions save time because there is no continuous auth checkhead for verification at this some time.

Answer:- doGet()	doPost()
1). This method is used to send the data in URL	This is used to send data in data packets
2). It follows the URL & query parameter mechanism for transmission.	It follows the request-response mechanism
3). It is less safe because data is easily accessible.	It is safe because encrypted data packets travel through network.

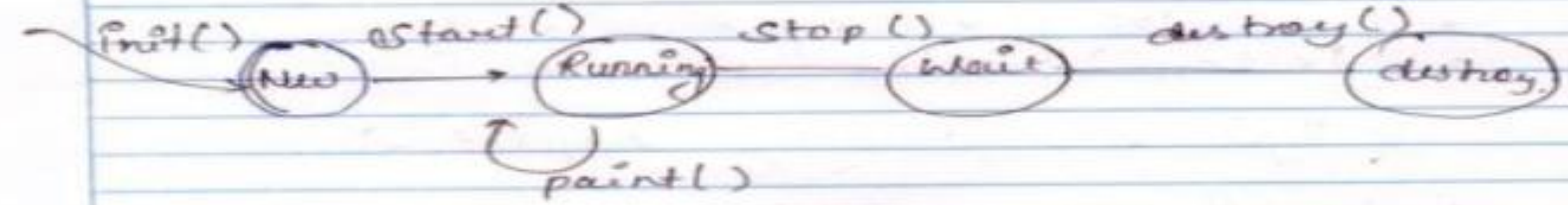
4).	It is assigned with Submit methods when data travells from one frontend technology to backend.	It is used when multiple servers or networks are interacting with each other.
5.	It is faster because there is no overhead.	It is slower because of weight of data & transportation overhead

Section-B

2.	Application	Applets
•	There are general Java codes that are made by programmers	There are a client-side technology codes that are used to create GUI & animations.
•	Java applications are used to write the business logic or circuit working mostly in jar.	Applets are used to present the frontend view or the interfaces of Java applications.

These are platform independent & do not require dependencies.	Applets can only be viewed with 1). Java-enabled web-browsers 2). Applet viewers.
• Simple Applications cannot be networked or transferred from one client to another.	Applets are a web-based technology that can be transferred over a network.
• Java Applications are pure Java codes and do not include any other technology.	Java Applets are a technology which is written inside HTML & is supported by HTML through <applet> </applet> tags.
• They are comparatively less secure than the applets because they have comparatively more access to resources.	Since they have limited access to resources they are considered more secure. Ex- Web applications

Life cycle of an applet



1. New
   - It is the first state of applet where the applet is initialized using the init() method.
   - Here all declaration & initialization occur.

2. Running
   - Running is the second state where the applet after the start() and paint() method comes.
   - Here the actual implementation of actual applet is given

3. stop / wait
   - When an applet has to wait for some event to occur it is waiting in stop phase
   - function for this is stop().
   - Whenever applet window is minimized.

(4) destroy

- When a applet is removed from main memory.
- When tab of the applet is closed.
- Destroy() method is called in this case.

Steps for creating Applets:

Step-1 Applets cannot be made without using the awt library so first step is to import the awt library & the event library.

```
import java.awt.*;
import java.awt.event.*;
import java.awt.Applet.*;
```

Step-2. Now we need to form the class that will have applet implemented.

- This class should extend Applet Base class.

```
class appletPractice extends Applets
{
}
```

Step-3. In the above class you need to override the necessary functions like init(), paint(), start(), stop() and destroy() according to your logic.

Step 4. We can also make applets using the graphics library that offer components for GUI

```
class public void paint (Graphics g)
{
    g.drawString("Hi");
}
```

Step-5 Now Save this file with some name with .java extention & add to the HTML file:

```
<applet code="applet_filename.java" width="100"
height="50"> </applet>.
```

Section-C

Q. Answer-7.

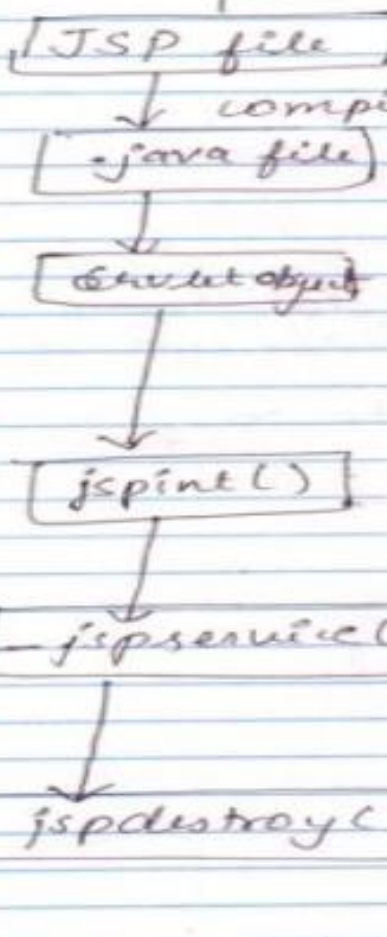
- (JSP Java Server Page)
- JSP stands for Java Server Page technology.
- It is a technology that is used to give dynamic content over a network in java.

- A JSP has more access to almost all Java APIS & Libraries therfore it can have a robust infrastructure for making applications.
- A JSP can be used to write designing & development both the logics in a structured manner that can be easily managed.
- No recompilation / redeployment needed - Since JSP is a dynamic technology no recompilations / redeployments after any changes are needed.
- The JSP files are made with extension - filename.jsp.

Lifecycle of JSP program

Step-1: Translation to servlet.
Step-2: Compiled & loaded the classes.
Step-3: jspInit()
Step-4: _jspService()
Step-5: jspDestroy()

A10



a .jsp file is translated to .java file & then performs further tasks.

here instantiation happens of the servlet object so that it can provide request-response mechanism

This method is used for initializations, declarations of the necessary components for a service.

This core method provides service to the calling programs in a seamless manner.

This method is used to end the jsp service & remove from the main memory.

- You are allowed to write both HTML & logic code inside the JSP files.
- Therefore there are 4 components of JSP.

JSP components

- Expression
- Scriptlets
- Directives
- Declarations

1. Expressions

- These are the regular expression combination of operators / variables that result to something.
- The expressions are converted to strings & then printed
- Syntax <%= //code. %>

2. Scriptlets

- These are the logical code that is stored inside the HTML files.
- They are mostly not presented on screen.
- Syntax - <% //code %>.

3. Directives

- Directives are used to import various packages inside the JSP file.
- syntax <%@ //code %>

4. (Declarative)

- These are used for declarations & definitions signatures of the necessary components of a Jsp file.
- Syntax: <%! //code. %>

(Why jsp over servlets)

(1) Servlets do not allow us to write both design & development logic in a single place but using jsp we can do so.

(2) JSP have access to almost all the Java API & libraries which makes it more suitable as now jsp is more robust & it supports an easier & library enabled development of applications which is a faster way.

(3) JSP do not need recompilation on redeployment after the process of updations but the servlets needed to be recompiled everytime after updations also we needed to redeploy the servlets to the server for error free & seamless working of servlets.

(4). In servlets we can only write the server side code but we can use JSP to write both the codes together which makes it easier & faster to manage.

(5). Some jsps are the extend extension of java servlets so a more better & advanced implementation of service() functions are provided in jsps which are not present in java servlets.

Example of JSP code.

```
<%@ page "import" = "java.util" %> //definitio
<html>
<body>
<%= (1+2) %> // expressions
<%! int counter = 0 %> // declaratives
<% out.println("Hi") %> // scriplets
<body>
<html>
```