(JAVA PROGRAMMING - 22412)

Unit I : Basic Syntactical constructs in JAVA.

(Weightage - 10 marks)

PRANJAL SAVEL (SY-COMPS)

1.1 : JAVA FEATURES & JAVA PROGRAMMING Environment

Questions

* Describe any four features of java. (4M)
* Explain any four features of JAVA. (4M)
* List any eight features of JAVA. (2M)

JAVA Features

(1) Simple / Simplicity

* JAVA is easy to write and more readable and eye catching.
* most of concepts are drawn from C++ thus making JAVA learning simpiler.
* If we understand concept of OOP the JAVA learning is easier and understandable.

(2) Object Oriented

* JAVA is object oriented programming language.
* Like C++, JAVA provides most of object oriented features.
* JAVA is pure OOPs language unlike OOP using C++ (it semi OOP).

(3) Secured

* JAVA is secured because no explicit pointer.
* Program runs inside virtual machine sandbox.
* Security becomes important issue for language that is used for programming on internet. So, JAVA enables construction of virus-free, tamper-free system.

(4) Platform independent

Java has implemented 'Write Once Run Anywhere' known as platform independent strategy.
Program written on one platform can be run on any other platform.
Changes and upgradation in O.S does not affect program at run.

(5) ~~Portable~~ Architectural Neutral

JAVA tech application compiled in one Architecture (hardware - RAM, Harddisk) and that compiled program runs on any hardware is called Architectural Neutral.

(6) Portable

In JAVA technology ~~are~~ the applications are compiled and executed in any O.S and any hardware hence we can say java is portable.

Linux ⇒ Window ⇒ MAC

(7) Robust

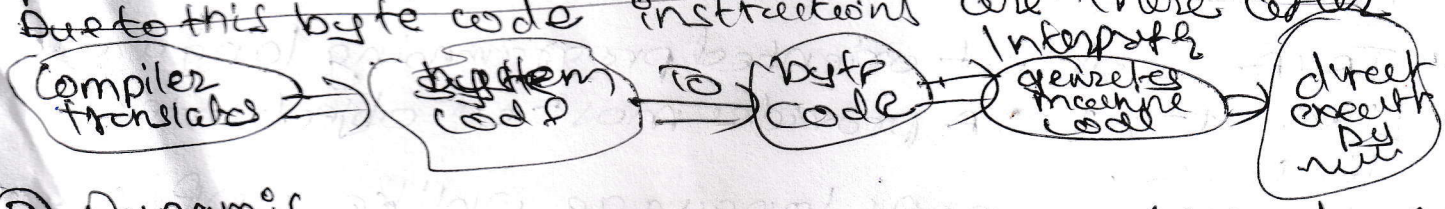
Java makes an effort to eliminate error prone situations by emphasizing mainly on compile time error checking and runtime checking. Any tech. is good at two main areas are said to be ROBUST. memory allocation & Exception Handling

(8) Distributed

Java was designed with distributed environment. JAVA can be transmit, run over internet.

(9) Compiled and Interpreted

Java is only language, which has compiles and interpreter both. This has been designed to ensure platform independence nature for a language. Due to this byte code instructions are there in files



(10) Dynamic

* It is more dynamic than C & C++ since it is designed to adapt to evolving environment.

(11) others ⇒ High performance, Multithreaded

Java Development Tools

- Javac ⇒ compiler that translates javacode file which (compiler) java interpreter can understand
- Java ⇒ java interpreter that run applets and applications (interpreter) interpreting byte code files.
- Javap (JAVA disassemble)
- ~~javap~~
- javah (for (header files)
- javadoc (HTML documents)
- jdb (JAVA debugger)

Object

* Objects are basic runtime entities. any real world Object can be treated as an object in OOP. Such as person, car etc.....
* An object is instance of class
* A entity that has state and behaviour is known as an object.
* Example, chair, bike, marker, pen, car etc.....
* It is physical or logical.

new keyword = used to allocate memory at runtime

program

* Define class employee with data members 'emp id', 'name' and 'salary'. Accept data for three objects and display it. (UM)

⇒
```
class employee
{
    int emp_id;
    string name;
    int salary;
    void getdata(int e, string n, int s)
    {
        emp_id=e;
        name=n;
        salary=s;
    }
    void display()
    {
        System.out.println(emp_id+" "+name+" "+salary);
    }
public static void main(String args[])
{
    employee E1=new employee;
    employee E2=new employee;
    employee E3=new employee;
    E1.getdata(101, "RITA", 5000);
    E2.getdata(102, "RAM", 6000);
    E3.getdata(103, "RAMU", 7000);
```

Simple JAVA Program

```
class simple    // declaring class
{
    public static void main(String args[])
    {
        System.out.println("Hello");
    }
}
```

1.2 Defining a class, creating object, accessing class members.

Class Questions

- Define class & object (2m) ***
- Define class employee with data members 'empid', 'name' and 'salary'. Accept data for three objects and display it. (Similar type)

class

A class is a group of object that has common properties. It is template or blueprint from which object are created

A class is a way of grouping objects having similar characteristics. It is user defined data type which represents on group of similar objects.

To create a class' object we need to specify data member & member function.

A class in java can contain => data member, method, contructor, block, class and interface.

Syntax

```
class <class_name>
{
    datamember;
    method;
}
```

Example

```
class student
{
    string name;
    int roll no;
    int age;
}
Student std = new student()
```

```
        E1.display();
        E2.display();
        E2.display();
    }
}
```

Output

```
101 RIYA 5000
102 RAM  6000
103 RAMU 7000
```

Program

Define a class circle having data members Pi and radius. Initialize and display values of data members also calculate area of circle and display also.

```
class circle
{
    float radius;      // Data members
    float pi=3.14;
    void getdata(float r)
    {
        radius=r;
    }
    void Area()
    {
        float area;
        area = radius*radius*pi;
        System.out.println("Radius = "+area);
    }
public static void main(String args[])
{
    circle c1=new circle();
    circle c2 = new circle();
    c1.getdata(2.14);
    c2.getdata(3.32);
    c1.Area();
    c2.Area();
}
}
```

1.2 Java Token and Data types, constants and Symbolic Constants, variables, dynamic initialization. data types, array and string, Scope of variable, type casting, and Standard defaults.

Question IMP

* Describe type casting in java with example. (1M)
* write all primitive data types available in JAVA with their storage size in bytes. (1M)
* Define array, State types (2M)

Java tokens

* Tokens are variable or various JAVA program elements which are identified by compiler.
* A token is smallest element of program that is meaningfeel to compiler.
* Token supported in JAVA include keywords, variables, contants, special charachters, operations etc.....
* Tokes are smallest unit of Program.

There are Five types of Token.

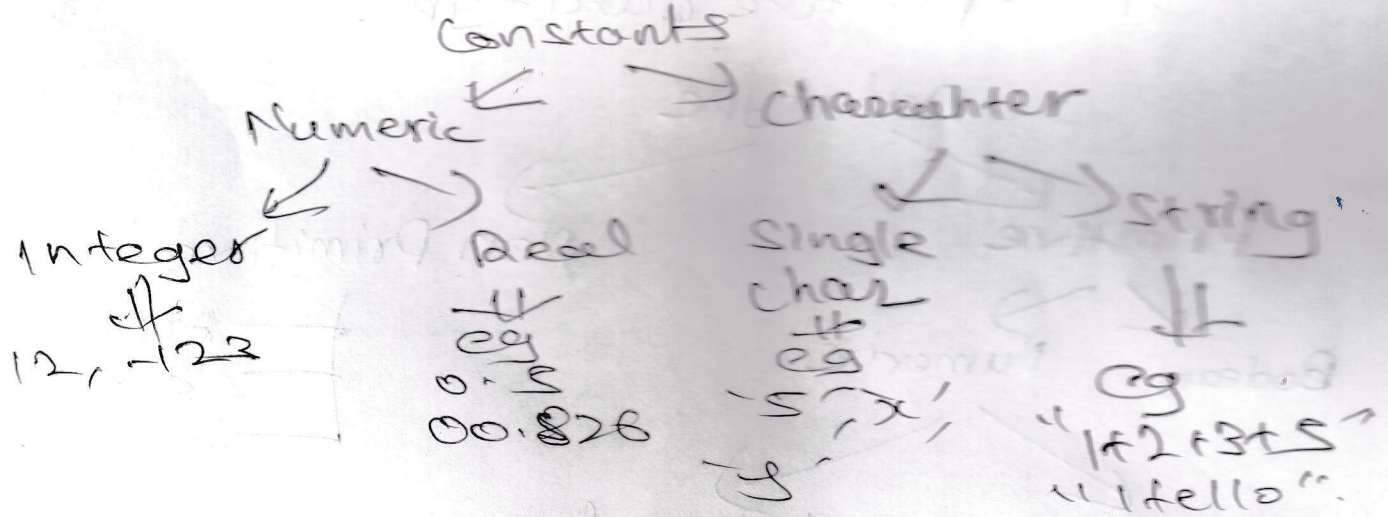
- Keywords
- Identifier
- Literals
- Operators
- Seperators.

keywords

abstract	continue	float	native	switch
assert	default	for	new	synchronized
boolean	double	goto	null	this
break	do	if	package	throw
byte	else	implements	private	throws
case	enum	import	protected	transient
catch	extends	instance of	public	true
char	false	int	returned	try
class	final	interface	short	void
const	finally	long	static	volatile
			strictfp	while.
			super	

Constants

Constants in JAVA are fixed values those does not chage during Execution of program JAVA support several type of constants those are



Space

\t ⇒ Tab
\b ⇒ BackSpace
\f ⇒ form
\n ⇒ line
\r = carriage
\" = Double quote
\' = Single quote.
\\ = Black slash.

Question : Scope of variable (2 marks)

variable is name of reserved area allocated in memory

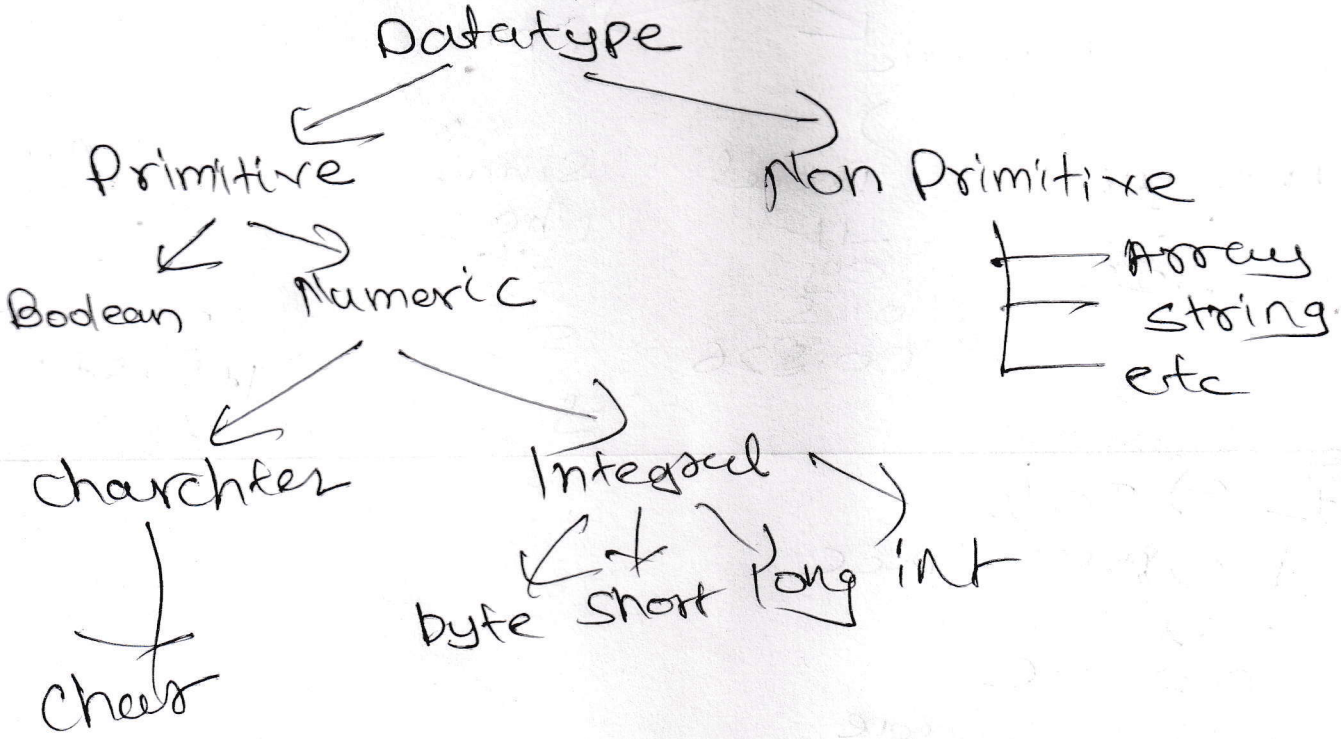
int data = 50 // data is variable

Types of variable

- Local → inside the method
- Instance → declared inside class outside method.
- Static → stable constant.

Datatypes in JAVA (2 marks or 4 marks)

In JAVA there are two Datatypes.
- Primitive datatype
- Non-primitive datatype.



Datatype	size
boolean	1bit
char	2byte
byte	1byte
short	2byte
int	4byte
long	8byte
float	4byte.
double	8byte

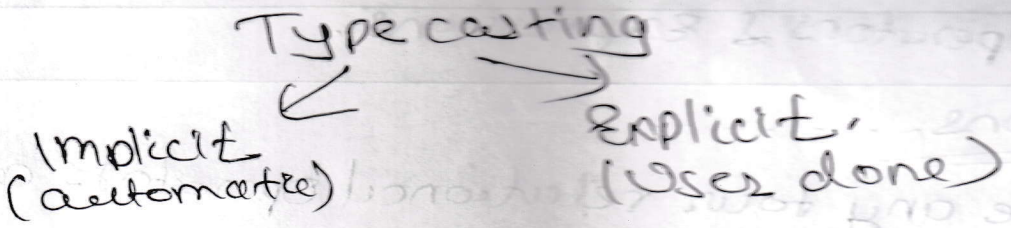
What is Array and it's type?

=> An array is group of like-typed variables that are reffered to common-name. Array of any type can be created and may have one or more dimensions.

types
- 1D
- 2D
- multidimensional.

Typecasting

- Typecasting means conversion of one datatype to another datatype.
- Example, float value can be converted into integer value.
- There are two types of type casting in JAVA.



Implicit

- In this the conversion is done automatically done by the compiler.
- In this smaller datatype is converted into larger datatype.
- Hierarchy is int → unsigned int → long → long → double → long double.
- In most operations, operands require to be of same type if they are not then compiler will convert them using above Hierarchy.
- when compiler tries to convert lower level data type to higher level data type then it is know as widening or Promotion.
- Higher ⇒ lower then narrowing or Coercion.

Example

```
double d = 2.8;
int i;
i = d;
S.O.P(i);
```

⇒ 2

Explicit

It refers to type conversion that is performed explicitly by user in program.
This can be performed by using two different forms.

* datatype (expression) → int (2.3);
* (datatype) expression → (int) 2.3;

Example - int (2.3);

In above example integer value the given value is converted into

O/P will be 2.

1.4 Operators & expressions.

Questions - - - - - -

Q// State any four relational operators and their use. (2m)

Q// Given syntax and example.
i) min()
ii) Sqrt()

Q// Explain switch case and condinal operator in java with suitable example. (3m)

Arithematic = +, -, *, /, %, ++, --

Increment/Decrement = ++ --

Relational = == != > < >= <=

Bitwise = & | ^=exor ~ =ex nor << =left shift >> =right shift. >>>> =shift right zero

Logical = && || !

Assignment operators

=	C = A + B
+=	C = C + A
-=	C = C - A
Samecall */ ^ |=	C = C operator A

C will assign A+B
C += A
C -= A
C operator = A

instance of operator

This operator is used only for object reference variable.

The operator checks whether the object is of particular type.

instance of operator is written as

Syntax (object reference variable) instanceof (class)

Eg boolean result = name instance of String

---

mathematical fun

min (values)
max(values)
abs(values)
floor(values)
sqr(values)
exp(values)
pow(values)

Syntax

math.operator(value)

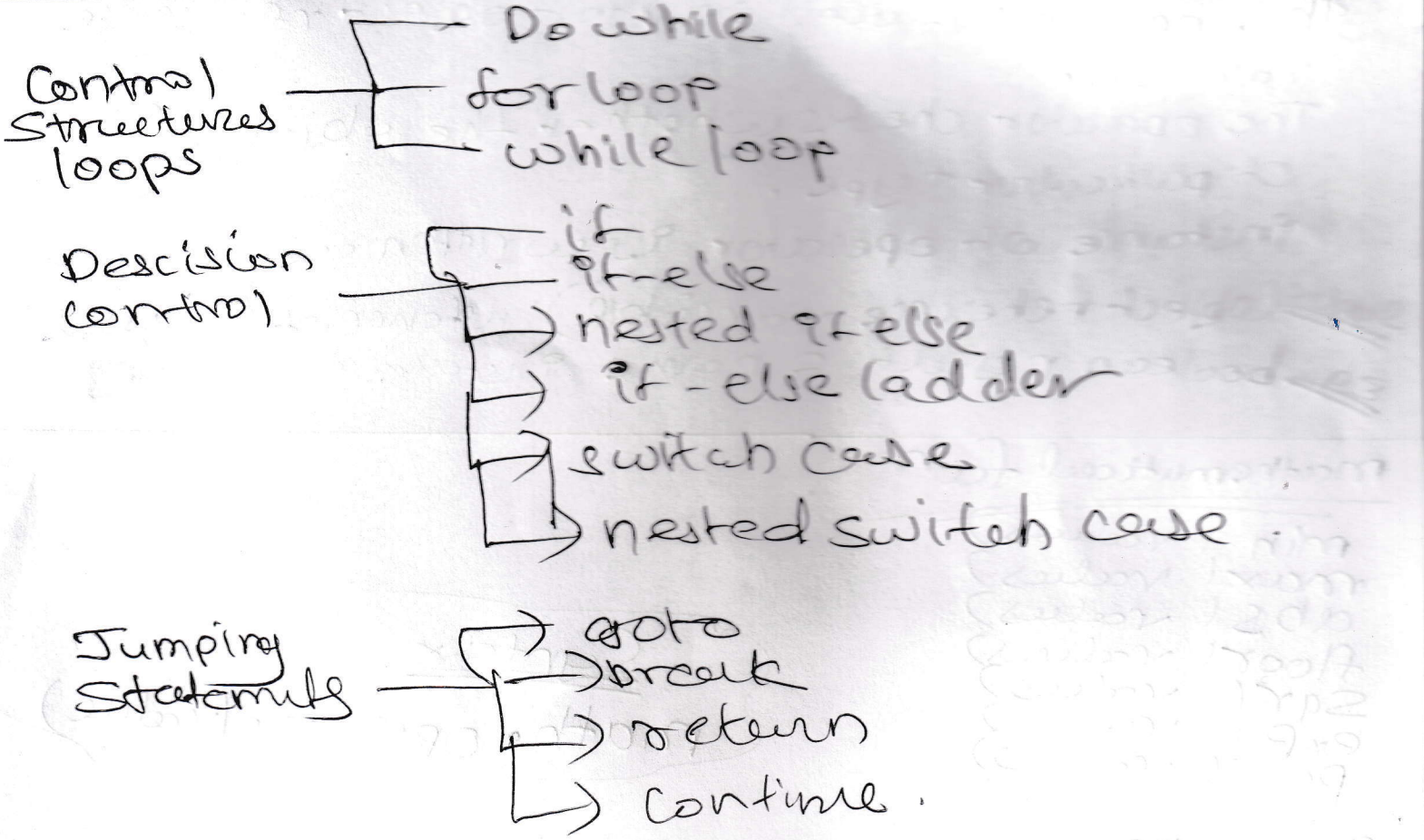
Program

```
class mathfun
{
  public static void main (String args[])
  {
     double val=25;
  System.out.println ("Square root of 25 "+math.sqrt(val));
  a=10, b=15;
  S.O.P( max(a,b)
  S.O.P(math.min(a,b)
  }
}
```

O/P

5
15
10

1.5 Decision making and looping



✱ Programs ✱

✱ Write a program to find reverse of number. (4 marks)

```
import java.util.*;
class Reverse
{
public static void main(String args[])
  {
    int n, rev=0, r;
System.out.println("Enter number");
Scanner y = new Scanner(System.in);
n = y.nextInt();
while(n!=0)
 {
  r= n%10;
 rev=(rev*10)+r;
  n=n/10;
 }
System.out.println("Reversed no= "+rev);
}
}
```

O/P
Enter number 1234
Reversed no= 4321

Factorial

```
class FactorialDemo {
  int num = 5;
  int fact = 1;
  public void factorial()
  {
    for(i=1;i<=num;i++)
      {
        fact = fact*i;
      }
    System.out.println("Number = " +num);
    System.out.println("Factorial = " +fact);
  }
}
JavaFactorial class
  {
public static void main(String args[])
    {
  FactorialDemo X = new FactorialDemo();
   X.factorial();
  }
}
```

O/P

```
Number = 5
Factorial = 120
```

& 1 1 1 1 1
2 2 2 2
3 3 3
4 4
5

→ Class Sample

```
{
public static void main (String args[])
{
 int i, j, a=1;
 for(i=5; i>=1 ;i--)
   {
     for(j=1;j<=i;j++)
        {
          System.out.println(a+" ");
        }
  System.out.println("\n");
   a++;
 }

}
```

Write a program to check whether the given number is prime or not.

```
import java.util.*;
class PrimeNo
{
 public static void main (String args[])
  {
  Scanner bin = new Scanner (System.in);
  System.out.println("Enter no= ");
   int num = bin.nextInt();
    int flag=0;
for(int i=2; i<num; ++i)
    {  if(num%i==0)
        {
       System.out.println(num+" is not PN");
       flag=1;
        break;
        }
    }
if(flag==0)
 {
System.out.println(num + " PN ");
 }
}
}
```

Armstrong Number or not.

```
class Armstrong
{
public static void main(String args[])
  {
    int n,c=0,a,d;
System.out.println("Enter no=");
Scanner sc= new Scanner(System.in);
  n= sc.nextInt();
   d=n;
  while(n>0)
    {
      a=n%10;
      n=n/10;
      c+=(a*a*a);  //c=c+(a*a*a)
    }
  if(d==c)
{
System.out.println("Armstrong number");
}
else
{
System.out.println("Not an Armstrong
                    number");

  }

}
```