

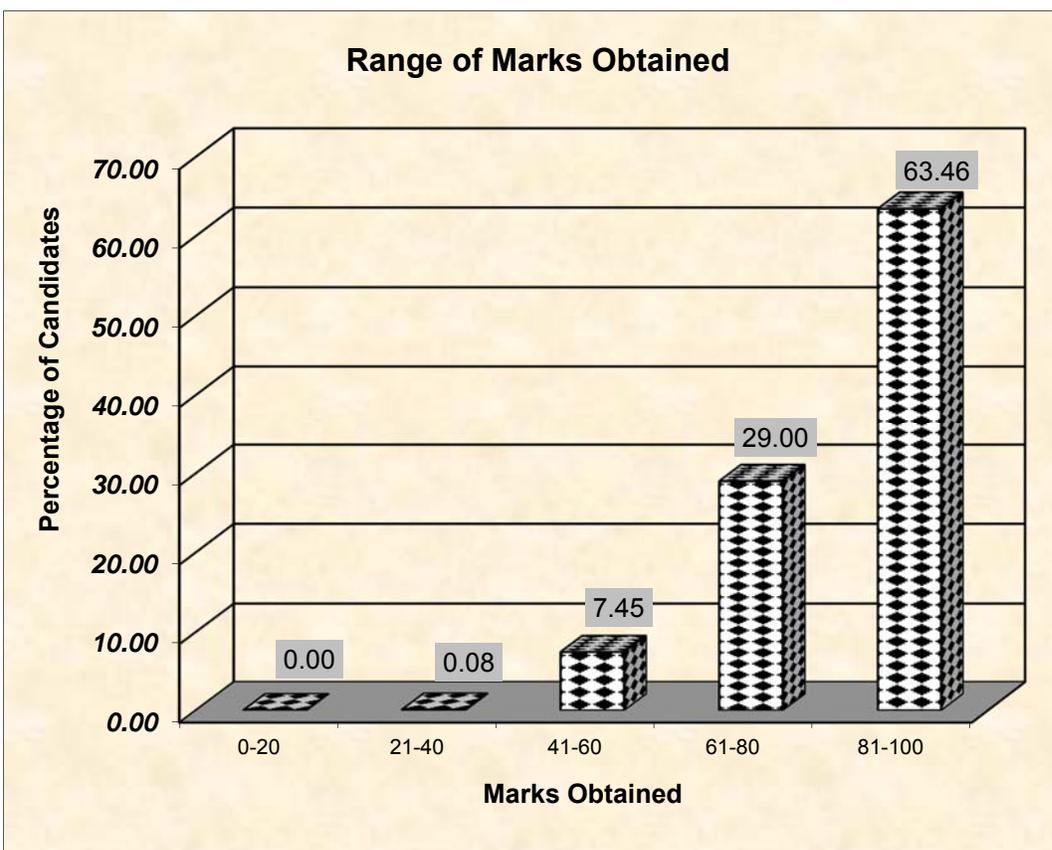
COMPUTER APPLICATIONS

STATISTICS AT A GLANCE

Total Number of students who took the examination	89,447
Highest Marks Obtained	100
Lowest Marks Obtained	18
Mean Marks Obtained	83.82

Percentage of Candidates according to marks obtained

Details	Mark Range				
	0-20	21-40	41-60	61-80	81-100
Number of Candidates	2	73	6665	25943	56764
Percentage of Candidates	0.00	0.08	7.45	29.00	63.46
Cumulative Number	2	75	6740	32683	89447
Cumulative Percentage	0.00	0.08	7.54	36.54	100.00



COMPUTER APPLICATIONS

ANALYSIS OF PERFORMANCE

Question 1

- (a) Which of the following are valid comments?
- (i)/* comment
*/
 - (ii)/* comment
(iii)//
comment
 - (iv)*/ comment */ [2]
- (b) What is meant by a package? Name any two Java Application Programming Interface packages. [2]
- (c) Name the primitive data type in Java that is:
- (i) a 64-bit integer and is used when you need a range of values wider than those provided by *int*.
 - (ii) a single 16-bit Unicode character whose default value is '\u0000'. [2]
- (d) State one difference between the floating point literals *float* and *double*. [2]
- (e) Find the errors in the given program segment and re-write the statements correctly to assign values to an integer array.
- ```
int a = new int(5);
for(int i=0;i<=5;i++) a[i]=i;
```
- [2]

### Examiners' Comments

- (a) Most candidates answered correctly. However a few were not clear about the starting and ending symbols of multi-line comment.
- (b) Most candidates answered this question correctly.
- (c) Most candidates answered the first part correctly while a very few answered the second part correctly.
- (d) Most candidates answered the question correctly.
- (e) A few candidates could not identify all the 3 errors while 2 or just 1 of the errors were identified by most candidates.

### Suggestions for teachers

- The Topic "Comments in Java" should be taught and revised thoroughly.
- Students must be trained to write programs with comments.
- The proper definition of the term "package" to be explained thoroughly followed by names of important packages.
- Students must be guided and reminded constantly to learn such topics thoroughly as they are examined in the compulsory section only.
- The "Data Type" topic should be covered thoroughly under different heads.
- Differences between all types of data types to be clearly discussed in class.
- Explain with conceptual clarity the Syntax of declaration of 1-D array and motivate the students to learn it thoroughly.
- Test expression of the loop to create 1-D array to be comprehensively explained.

### **MARKING SCHEME**

#### **Question – 1**

- (a) (i)  
(iii)
- (b) A package is a group of related classes  
Any two of the following:

|               |                     |                   |
|---------------|---------------------|-------------------|
| java.applet   | java.sql            | javax.sound       |
| java.awt      | java.text           | javax.swing       |
| java.beans    | java.util           | javax.transaction |
| java.io       | javax.accessibility | javax.xml         |
| java.lang     | javax.crypto        | org.ietf          |
| java.math     | javax.imageio       | org.omg           |
| java.net      | javax.naming        | org.w3c           |
| java.nio      | javax.net           | org.xml           |
| java.rmi      | javax.print         |                   |
| java.security | javax.rmi           |                   |

- (c) (i) long  
(ii) char

- (d) The float data type is a single-precision 32-bit (or 4 bytes) OR Default value 0.0f OR 8 digit precision  
 The double data type is a double-precision 64-bit (or 8 bytes) OR Default value 0.0d OR 16 digit precision
- (e) `int a[]=new int[5];` OR `int []a=new int[5];`  
`(for(int i=0;i<5;i++))` OR `for(int i=0;i<=4;i++)`

## Question 2

- (a) Operators with higher precedence are evaluated before operators with relatively lower precedence. Arrange the operators given below in order of higher precedence to lower precedence.
- (i) `&&` (ii) `%` (iii) `>=` (iv) `++` [2]
- (b) Identify the statements listed below as assignment, increment, method invocation or object creation statements.
- (i) `System.out.println("Java");`  
 (ii) `costPrice = 457.50;`  
 (iii) `Car hybrid = new Car();`  
 (iv) `petrolPrice++;` [2]
- (c) Give two differences between the *switch* statement and the *if-else* statement. [2]
- (d) What is an infinite loop? Write an infinite loop statement. [2]
- (e) What is a constructor? When is it invoked? [2]

Examiners' Comments:

- (a) Most candidates who were clear about precedence of operators wrote the correct answer.
- (b) Some candidates appeared confused between the method invocation (i) and object creation (ii).
- (c) Answered correctly by candidates who thoroughly studied the theoretical notes on if-else and switch-case.
- (d) The first part was answered correctly by most candidates. Examples written however were incorrect.
- (e) The definition of a constructor was answered correctly though some candidates did not cover all the points in their definition.

Suggestions for teachers

- Hierarchy/precedence/order of all types of operators should be taught thoroughly with adequate evaluation exercises based on precedence.
- Assign practice exercises so that students are able to identify different types of statements.
- Provide concise notes covering all possible questions on if-else and switch-case and explain the role of each keyword like else, default, case, etc.
- The topic “Variations of loops covering, multiple initialization expressions, multiple update expressions and empty loop should be explained in class with proper examples and exercises.
- Invoking of a constructor along with object creation needs to be explained in detail.

**MARKING SCHEME**

**Question - 2**

- (a) (iv), (ii), (iii), (i) All 4 correct  
Only 1 incorrect
- (b) (i) method invocation  
(ii) assignment  
(iii) object creation  
(iv) increment All 4 correct  
Only 1 incorrect
- (c) switch:  
can perform only equality (= =) comparison OR  
can have a number of possible execution paths OR  
works with the byte/short/char/int primitive data types OR  
multiple branching OR  
is used along with case statement OR default statement  
If-else:  
Can perform all relational comparisons (<, >, >=, <=, =, !=) OR  
executes a certain section of code *only if* a particular test evaluates to true OR  
provides a secondary path of execution when an "if" clause evaluates to false OR  
works with any data type OR conditional/control flow/decision statements
- (d) A sequence of instructions which loops/iterates/repeats endlessly.  
Any valid example which shows:  
the loop having no terminating condition OR

having a condition that can never be met OR  
one that causes the loop to start over  
for(;;) or while(true){}

- (e) A constructor is a member function or method of a class that has the same name as the class OR has no return type not even void. OR initializes data members of a class.

It is invoked at the time of object creation

### Question 3

- (a) List the variables from those given below that are composite data types.

(i) static int x; (iv) boolean b;

(ii) arr[i]=10; (v) private char chr;

(iii) obj.display(); (vi) String str;

[2]

- (b) State the output of the following program segment:

```
String str1= "great"; String str2= "minds";
```

```
System.out.println(str1.substring(0,2).concat(str2.substring(1)));
```

```
System.out.println(("WH"+(str1.substring(2).toUpperCase())));
```

[2]

- (c) What are the final values stored in variables x and y below?

```
double a = - 6.35;
```

```
double b = 14.74;
```

```
double x = Math.abs(Math.ceil(a));
```

```
double y = Math rint(Math.max(a,b));
```

[2]

- (d) Rewrite the following program segment using *if-else* statements instead of the ternary operator.

```
String grade=(mark>=90) ? "A" : (mark>=80) ? "B" : "C";
```

[2]

- (e) Give the output of the following method:

```
public static void main(String[] args){
```

```
int a = 5;
```

```
a++;
```

```
System.out.println(a);
```

```
a--=(a--) - (--a);
```

```
System.out.println(a); }
```

[2]

- (f) What is the data type returned by the library functions:

- (i) compareTo()  
(ii) equals() [2]
- (g) State the value of **characteristic** and **mantissa** when the following code is executed.  
String s = "4.3756";  
int n = s.indexOf('.');  
int characteristic=Integer.parseInt(s.substring(0,n));  
int mantissa=Integer.valueOf(s.substring(n+1)); [2]
- (h) Study the method and answer the given questions.  

```
public void sampleMethod()
{
 for(int i=0;i<3;i++)
 {
 for(int j=0;j<2;j++)
 {int number = (int)(Math.random() * 10);
 System.out.println(number);
 }
 }
}
```
- (i) How many times does the loop execute?  
(ii) What is the range of possible values stored in the variable *number*? [2]
- (i) Consider the following class:  

```
public class myClass {
 public static int x=3, y=4;
 public int a=2, b = 3; }
}
```
- (i) Name the variables for which each object of the class will have its own distinct copy.  
(ii) Name the variables that are common to all objects of the class. [2]
- (j) What will be the output when the following code segments are executed?  
(i) String s="1001";  
int x=Integer.valueOf(s);  
double y=Double.valueOf(s);  
System.out.println("x="+x);  
System.out.println("y="+y);  
(ii) System.out.println("The king said \"Begin at the beginning!\" to me."); [2]

### Examiners' Comments

- (a) Some candidates wrote all three options correctly but most could identify one or two options correctly.
- (b) A few candidates who were clear about string functions wrote the correct answer.
- (c) Some candidates were confused regarding the sign (+ or -) in the answer and whether the answer should be of double type or int.
- (d) Most candidates answered correctly. Some candidates, however, printed the grades instead of assigning them.
- (e) Some candidates were not clear about evaluation of arithmetic assignment operators after increment/decrement operators.
- (f) A few candidates wrote literals as answers as they failed to read the question carefully and not the data types as asked. Some candidates were not clear about the data type int and wrote the full word integer.
- (g) Most candidates who had conceptual clarity of multiple functions wrote the correct answer.
- (h) Some candidates were not clear about random ( ) function and some appeared confused about nested loop.
- (i) Most candidates lacked the concepts based on static and instance variables.
- (j) Some candidates were unaware of value Of ( ) function to convert strings to primitive data types besides parsing. The role of escape sequences in outputs were not clear to some candidates.

### Suggestions for teachers

- Explain the difference between primitive and composite data types and give numerous examples to identify variables of each type.
- Practice questions on multiple string functions must be given in class regularly.
- Practice exercises on evaluation of the multiple and single mathematical functions must be given in class and students should be guided to revise and practice the same exercises for the exams.
- Practice questions on conversion of nested ternary operator expressions to if else and visa versa must be given in class.
- Order of evaluation of expressions using arithmetic assignment operators and increment/decrement operators must be clearly explained in class along with examples.
- When teaching string functions proper syntax of each function prototype (return type and type and number of parameters) should be explained and students should be made to note them in their registers. Students should also be advised to read the questions carefully at least twice before answering them.
- Functions form a very important part of the Java language hence a thorough explanation and additional practice questions on functions must be given.
- All library functions including random( ) must be tried as output questions, working practically on the computer along with evaluation exercises theoretically. Nested loop concept – outer loop takes the next iteration only when the entire inner loop goes through all its iterations once – should be explained with additional nested loop examples.

Suggestions for teachers

- The keyword “static” and its use must be explained with examples.
- ValueOf( ) function as a substitute for parsing strings to primitive types must be taught with examples. Escape sequences and their use along with the list of important escape sequences must be taught in class.

**MARKING SCHEME**

**Question - 3**

- (a) (ii) arr (iii) obj (vi) str All 3 correct  
Any 2 correct
- (b) grinds  
WHEAT
- (c) 6.0  
15.0
- (d) if(mark>=90) grade="A";  
else if(mark>=80) grade="B";  
else grade="C"; All 3 correct  
Any 2 correct
- (e) 6  
6-(6-4)=4
- (f) (i) int  
(ii) boolean
- (g) 4  
3756
- (h) (i) 6 times  
(ii) 0 to 9 OR {1,2,3,4,5,6,7,8,9}
- (i) (i) a,b  
(ii) x,y
- (j) (i) x=1001  
y=1001.0  
(ii) The king said “Begin at the beginning!”to me.nextLine()

#### Question 4

Define a class named **movieMagic** with the following description:

Instance variables/data members:

- int year - to store the year of release of a movie
- String title - to store the title of the movie
- float rating - to store the popularity rating of the movie  
(minimum rating=0.0 and maximum rating=5.0)

Member methods:

- (i) movieMagic() Default constructor to initialize numeric data members to 0 and String data member to "".
- (ii) void accept() To input and store year, title and rating.
- (iii) void display() To display the title of a movie and a message based on the rating as per the table below.

| Rating     | Message to be displayed |
|------------|-------------------------|
| 0.0 to 2.0 | Flop                    |
| 2.1 to 3.4 | Semi-hit                |
| 3.5 to 4.5 | Hit                     |
| 4.6 to 5.0 | Super Hit               |

Write a main method to create an object of the class and call the above member methods. [15]

### Examiners' Comments

- (i) The Class name written by most candidates was different from the one asked in the question.
- (ii) Constructor syntax was incorrect (void used, variables declared inside constructor and initialized).
- (iii) if-else range > and < placed incorrectly.
- (iv) There was confusion between mathematical symbols  $\leq$ ,  $\geq$  and Java relational operators `<=`, `>=`.
- (v) Multiple functions were not written as asked.
- (vi) There was confusion between global and local variables.

### Suggestions for teachers

- Guide and advise students to carefully read the question and understand its requirements.
- Constructor syntax to be explained comprehensively with additional revision and practice on writing constructors.
- Guide and instruct students to practice to convert English statements having conditions with ranges to Java if-else using proper relational operators.
- Advise students in home exams against using  $\leq$ ,  $\geq$  instead of `<=`, `>=`.
- Programs on multiple functions of all types to be revised on a consistent basis in class.
- Difference between global and local variables and how to use them in multiple function programs should be explained thoroughly in class.

## **MARKING SCHEME**

### **Question - 4**

```
import java.io.*; //import java.util.*;
public class movieMagic
{
 int year; float rating; String title;

 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 //Scanner sc=new Scanner(System.in);

 public movieMagic()
 {
 year=0;
 rating=0.0f;
 title="";
 }

 public void accept()throws IOException//throws InputMismatchException
 {
 System.out.println("Enter year of release");
 year=Integer.parseInt(br.readLine()); // sc.nextInt();
 System.out.println("Enter title");
```

```

 title=br.readLine(); // sc.next();

 do
 {
 System.out.println("Enter rating (minimum 0.0 and maximum 5.0)");
 rating=Float.parseFloat(br.readLine()); // sc.nextInt();
 }
 while (!(rating>=0.0f && rating<=5.0.f));OR while (rating < 0.0f || rating
>5.0f);
 }

 public void display()
 {
 System.out.print("Title:"+title+" Rating: ");
 if (rating <=2.0) System.out.println("Flop");

 else if (rating<=3.4) System.out.println("Semi
Hit");

 else if (rating<=4.5) System.out.println("Hit");

 else System.out.println("Super Hit");
 }

 public static void main()throws IOException
 {
 movieMagic object = new movieMagic();

 object.accept();
 object.display();
 }
}

```

| <i>Step</i>                                                       |
|-------------------------------------------------------------------|
| Declaration of class and instance variables                       |
| Creating object of class BufferedReader/Scanner                   |
| Constructor properly declared and data members initialised        |
| accept() method declaration (with exception handling if required) |
| 3 Inputs correct                                                  |
| Output title in display() method                                  |
| Decision for rating<=2.0 and output                               |

|                                               |  |
|-----------------------------------------------|--|
| Decision for rating $\leq$ 3.4 and output     |  |
| Decision for rating $\leq$ 4.5 and output     |  |
| Decision for rating $>$ 4.5 and output        |  |
| main method declaration                       |  |
| Creation of object                            |  |
| 2 methods called correctly                    |  |
| Description of variables/ comments/ mnemonics |  |

**Question 5**

A special two-digit number is such that when the sum of its digits is added to the product of its digits, the result is equal to the original two-digit number.

Example: Consider the number 59.

Sum of digits =  $5 + 9 = 14$

Product of its digits =  $5 \times 9 = 45$

Sum of the sum of digits and product of digits =  $14 + 45 = 59$

Write a program to accept a two-digit number. Add the sum of its digits to the product of its digits. If the value is equal to the number input, output the message “Special 2-digit number” otherwise, output the message “Not a special 2-digit number”.

[15]

Examiners’ Comments:

Most candidates answered this question, but the following errors were commonly observed:

- (i) Checking for 2-digit number was missing.
- (ii) Extraction of digits using loop was not clear.
- (iii) Candidates were confused in the use of temporary variable for copying the original value in extraction of digits.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><u>Suggestions for teachers</u></p> <ul style="list-style-type: none"> <li>- How to validate data should be taught with the help of examples.</li> <li>- Extraction of digits logic <math>n\% = 10</math> and <math>n/=10</math> needs to be explained with numerous examples and exercises.</li> <li>- Why a copy of original value must be maintained should be explained with a dry run and which one to use where – if the original is used in extraction loop then a copy to be used in comparison or vice versa.</li> </ul> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## MARKING SCHEME

### Question - 5

```
public class special
{
 public void sampleMethod(int num)
 {
 int digit, sumDigit=0, prodDigit=1,
sum=0;
 int n=num;
 if(num>=10 && num<=99)
 { while(n>0)
 { digit=n%10;
 sumDigit+=digit;
 prodDigit*=digit;
 n/=10;
 }
 sum+=sumDigit+prodDigit;
 if(sum==num)
 System.out.println("Special
number");
 else
 System.out.println("Not a special number");
 }
 }
}
```

| Step                                          |
|-----------------------------------------------|
| Input number                                  |
| Initialise 2 sums and product                 |
| Store number in second variable               |
| Check if number is 2-digit                    |
| condition for number >0 while(n>0)            |
| Extract digit from number                     |
| Compute sum of digits                         |
| Compute product of digits                     |
| number/=10                                    |
| Compute total of sum and product of digits    |
| Check if total equals number                  |
| Output "Special Number" message               |
| Output "Not Special Number" message           |
| Description of variables/ comments/ mnemonics |

**OR**

```
public class alternateSpecial
{
 public void sampleMethod(String s)
 {
 int len,digit1, digit2, sumDigit, prodDigit, sum; String s1, s2;
 int n=Integer.parseInt(s);
 len=s.length();
 if (len==2) // if(n>=10 && n<=99)
 {
 s1=s.substring(0,1);
 s2=s.substring(1,2);

 digit1=Integer.parseInt(s1);
 digit2=Integer.parseInt(s2);

 sumDigit=digit1+digit2;
 prodDigit=digit1*digit2;

 sum=sumDigit+prodDigit;

 if(sum==n)
 System.out.println("Special Number");
 else
 System.out.println("Not a Special Number");
 }
 }
}
```

| Step                                        |
|---------------------------------------------|
| Input number as String                      |
| Convert String to integer                   |
| Check if number is 2-digit                  |
| Extract first digit                         |
| Extract second digit                        |
| Convert first digit from String to integer  |
| Convert second digit from String to integer |
| Compute sum of digits                       |
| Compute product of digits                   |
| Compute total of sum and product of digits  |
| Check if total equals number                |
| Output "Special Number" message             |

|                                               |  |
|-----------------------------------------------|--|
| Output "Not Special Number" message           |  |
| Description of variables/ comments/ mnemonics |  |

### Question 6

Write a program to assign a full path and file name as given below. Using library functions, extract and output the file path, file name and file extension separately as shown.

Input C:\Users\admin\Pictures\flower.jpg

Output Path: C:\Users\admin\Pictures\

File name: flower

Extension: jpg

[15]

#### Examiners' comments:

A direct question on strings but only a few candidates whose concepts were clear about string function "substring" answered it correctly. A few candidates confused this question to be of "files".

#### Suggestions for teachers

- Students need to be guided and instructed on how to read each question at least twice especially programming questions. The function "substring" is an important and useful function as it makes any logic of string program very simple, short and easy. Hence, this function should be taught and explained in detail with examples and programs using substring in class.

### MARKING SCHEME

#### Question - 6

```
public class stringHandling
{
 public void sampleMethod()
 {
 String s="C:\\users\\admin\\pictures\\flower.jpg";
 int len,i, pos=0; char c; String fpath="", fname="", fextn="";
 len=s.length();
 for(i=len-1;i>=0;i--)
 {
 c=s.charAt(i);
 if(c=='.')
 {
 fextn=s.substring(i+1);
 pos=i;
 }
 if(c=='\\')
 {
 fname=s.substring(i+1,pos);
 }
 }
 }
}
```

```

 fpath =
s.substring(0,i+1);
 break;
 }
}
System.out.println("Path: "+fpath);
System.out.println("File name: "+fname);
System.out.println("Extension: "+fextn);
}
}

```

| Step                                                                    |
|-------------------------------------------------------------------------|
| Assign value to String (accept single slash instead of double slash)    |
| Declare variables (ignore initialisation)                               |
| Find length of string                                                   |
| Loop                                                                    |
| Extract character                                                       |
| Check if character is '.'                                               |
| Extract substring for file extension                                    |
| store position of '.'                                                   |
| Check if character is '/' (accept single slash instead of double slash) |
| Extract substring for file name                                         |
| Extract substring for file path                                         |
| break to exit loop                                                      |
| Output file path, file name, file extension                             |
| Description of variables/ comments/ mnemonics                           |

**OR**

```
public class alternateSolution
```

```

{
 public void sampleMethod()
{
 String s="C:\\users\\admin\\pictures\\flower.jpg";

 int pos1=0, pos2=0; String fpath="", fname="", fextn="";

 pos1=s.lastIndexOf("\\");

 pos2=s.indexOf('.');

 fpath=s.substring(0,pos1+1);

 fname=s.substring(pos1+1, pos2);

```

```

 fextn=s.substring(pos2+1);
System.out.println("Path: "+fpath);
System.out.println("File name: "+fname);
System.out.println("Extension: "+fextn);
}
}

```

| Step                                                                   |
|------------------------------------------------------------------------|
| Assign value to String (accept single slash instead of double slash)   |
| Declare variables (ignore initialisation)                              |
| Find last index of slash (accept single slash instead of double slash) |
| Find index of '.'                                                      |
| Extract substring for file path                                        |
| Extract substring for file name                                        |
| Extract substring for file extension                                   |
| Output file path                                                       |
| Output file name                                                       |
| Output file extension                                                  |
| Description of variables/ comments/ mnemonics                          |

### Question 7

Design a class to overload a function area() as follows:

- (i) double area(double  $a$ , double  $b$ , double  $c$ ) with three double arguments, returns the area of a scalene triangle using the formula:

$$\text{area} = \sqrt{s(s-a)(s-b)(s-c)}$$

$$\text{where } s = \frac{a+b+c}{2}$$

- (ii) double area(int  $a$ , int  $b$ , int  $height$ ) with three integer arguments, returns the area of a trapezium using the formula:

$$\text{area} = \frac{1}{2} \text{height}(a+b)$$

- (iii) double area(double  $diagonal1$ , double  $diagonal2$ ) with two double arguments, returns the area of a rhombus using the formula:

$$\text{area} = \frac{1}{2} (\text{diagonal1} \times \text{diagonal2})$$

[15]

### Examiners' Comments

- (i) Most candidates appeared confused in the return type syntax – both void and double used together. If void used then instead of System.out.println( ) return used and if double used then no return but System.out. println( ) used.
- (ii) Mathematical expressions conversion to Java expressions incorrect -  $\frac{a+b+c}{2}$ .

### Suggestions for teachers

- Function prototype syntax with overloading to be discussed thoroughly.
- At least 10 – 12 programs on function overloading covering all logics must be done in class and ensure that all students have these programs in their registers for the purpose of revision during exams.
- Students should be made to practice to convert mathematical formulae to Java Expression.

## **MARKING SCHEME**

### **Question - 7**

```
public class overload
{
 double area=0;

 public double area(double a, double b, double c)
 {
 double s=(a+b+c)/2;
 area=Math.sqrt(s*(s-a)*(s-b)*(s-c));
 return area;
 }
 public double area(int a, int b, int height)
 {
 area=0.5*height*(a+b);
 return area;
 }
 public double area(double diagonal1, double diagonal2)
 {
 double area=0.5*(diagonal1 *diagonal2);
 return area;
 }
}
```

| Step                                               |
|----------------------------------------------------|
| Class name different from method name (area)       |
| Declare variable for area                          |
| First method declaration                           |
| Computing semi-perimeter, s                        |
| Computing area of triangle : $s*(s-a)*(s-b)*(s-c)$ |
| Computing area of triangle : Math.sqrt()           |
| return area                                        |
| Second method declaration                          |
| Computing area of trapezium                        |
| return area                                        |
| Third method declaration                           |
| Computing area of rhombus                          |
| return area                                        |
| Description of variables/ comments/ mnemonics      |

### Question 8

Using the switch statement, write a menu driven program to calculate the maturity amount of a Bank Deposit.

The user is given the following options:

- (i) Term Deposit
- (ii) Recurring Deposit

For option (i) accept principal(P), rate of interest( $r$ ) and time period in years( $n$ ). Calculate

and output the maturity amount(A) receivable using the formula  $A = P \left[ 1 + \frac{r}{100} \right]^n$

For option (ii) accept Monthly Installment (P), rate of interest( $r$ ) and time period in months ( $n$ ). Calculate and output the maturity amount(A) receivable using the formula

$$A = P \times n + P \times \frac{n(n+1)}{2} \times \frac{r}{100} \times \frac{1}{12}$$

For an incorrect option, an appropriate error message should be displayed.

[15]

### Examiners' comments

Common mistakes:

- (i) Mathematical expressions converted to Java incorrectly.
- (ii) Menu not displayed or choice not input.
- (iii) Break missing at the end of each case or default not present.
- (iv) Declaration of variables was not properly done.

### Suggestions for teachers

- Regular practice exercises must be undertaken in class to convert algebraic expressions to Java.
- The importance and use of displaying menu or inputting choice to be stressed upon.
- Syntax of switch-case to be thoroughly discussed and revised in connection with lot of programming questions.
- How to properly place declaration of variables storing data as per requirement of the question must be reinforced.

## **MARKING SCHEME**

### **Question 8**

```
import java.io.*; //import java.util.*;
public class bank
{ public void sampleMethod()throws IOException // throwsInputMismatchException
 { double P,A=0,r,n,x; int choice;
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 //Scanner sc=new Scanner(System.in);
 System.out.println("Enter Choice(1) Term Deposit (2) Recurring Deposit ");
 choice=Integer.parseInt(br.readLine());//sc.nextInt();
 switch(choice)
 {
 case 1:
 System.out.println(" Enter Principal:");
 P=Double.parseDouble(br.readLine());//sc.nextDouble();
 System.out.println(" Enter rate of interest:"); r=Double.parseDouble(br.readLine());
 System.out.println(" Enter time period in years:");
 n=Double.parseDouble(br.readLine());//sc.nextDouble();
 x=1.0+r/100.0;
 A=P*(Math.pow(x,n));
 break;
 case 2:
 System.out.println(" Enter Monthly Instalment:");
 P=Double.parseDouble(br.readLine());//sc.nextDouble();
 System.out.println(" Enter rate of interest:"); r=Double.parseDouble(br.readLine());
 System.out.println(" Enter time period in months:");
 n=Double.parseDouble(br.readLine());//sc.nextDouble();
 x=P*n;
 A=x+ P*(n*(n+1)/2.0)*(r/100.0)*(1.0/12.0);
```

```

 break;
 default: System.out.println("Invalid input");
} System.out.print("Amount = Rs."); System.out.printf("%.2f",A); } }
OR Sytem.out.println("Amount = Rs." +A)

```

| Step                                                       |
|------------------------------------------------------------|
| Output menu                                                |
| Input option                                               |
| switch statement                                           |
| case 1 and break                                           |
| Input 3 parameters (any data type)                         |
| Compute Amount for term deposit (ignore data types)        |
| case 2 and break                                           |
| Input 3 parameters (any data type)                         |
| Computing Amount for recurring deposit (ignore data types) |
| default statement with appropriate message                 |
| Output Amount (ignore formatting)                          |
| Description of variables/ comments/ mnemonics              |

### Question 9

Write a program to accept the year of graduation from school as an integer value from the user. Using the Binary Search technique on the sorted array of integers given below, output the message “Record exists” if the value input is located in the array. If not, output the message “Record does not exist”.

{1982, 1987, 1993, 1996, 1999, 2003, 2006, 2007, 2009, 2010}

[15]

Examiners' comments:

- (i) "Binary Search" is a very direct and fixed logic hence question based on it has to be a full mark question. However some candidates were not clear about the logic or its working. Sequence of statements were incorrect thus giving the impression that the statements were learnt thought rote memory, for e.g.:
- Mid value calculated outside while loop OR instead of comparing the search value with the value at mid position the comparison is done between search value and mid position i.e., instead of arr[mid] just mid is used lower = mid + 1 and upper = mid – 1 confused or interchanged according to the conditions.
- (ii) Syntax of initialization of 1-D array whether thru input or direct were not correct.
- (iii) Outputting of messages incorrectly placed.

Suggestions for teachers

- BINARY SEARCH technique should be taught with proper dry runs with different data (numeric and string). Each statement / step should be taught with a dry run with the data and then explained. Further students should be given plenty of exercises to dry run data with this technique and see if the correct result of binary search is achieved.
- Syntax of array initialization to be discussed and students should be given practice exercises to declare 1-D arrays.
- Dry run of this logic very essential in explaining to students.

**MARKING SCHEME**

**Question 9**

```
import java.io.*; //import java.util.*;
public class BinarySearch {
 public static void main() throws IOException
 { int[] intArray = {1982, 1987, 1993, 1996, 1999, 2003, 2006, 2007, 2009, 2010};
 int searchValue = 0;
 BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
 //Scanner sc=new Scanner(System.in);
 System.out.print("Enter a number to search for: ");
 searchValue = Integer.parseInt(br.readLine());//sc.nextInt();
 boolean b=false;
 int start, end, mid;
 start = 0;
 end = intArray.length - 1; // end=9
 while (start <= end) {
 mid = (start + end) / 2;
 if (intArray[mid] == searchValue)
 { System.out.println("Record exists"); b=true; break;}
 else if (intArray[mid] < searchValue)
 start = mid + 1;
 }
 }
}
```

1 mark

```

else
 end = mid - 1;
}
if (b==false) System.out.println("Record does not exist");
}
}

```

| Step                                                                                                                                                                      |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Assign (or accept) values to integer array                                                                                                                                |
| Input number to be searched                                                                                                                                               |
| Initialize boolean or integer value to check if search number exists in array (Deduct mark if boolean/integer value status does not change when search number is located) |
| Initialise start index and end index of array                                                                                                                             |
| Condition statement while(start<=end)                                                                                                                                     |
| Compute middle index                                                                                                                                                      |
| Check if array item at middle index equals search number                                                                                                                  |
| Output "Record exists", break to exit loop                                                                                                                                |
| Check if array item at middle index less than search number                                                                                                               |
| start index = middle index +1                                                                                                                                             |
| Check if array item at middle index greater than search number                                                                                                            |
| end index = middle index - 1                                                                                                                                              |
| Condition and output message "Record does not exist"                                                                                                                      |
| Description of variables/ comments/ mnemonics                                                                                                                             |

### **Topics /Concepts found difficult**

- Multi-line comments, escape sequences, default values of data types.
- Precedence of operators.
- Identifying variables of different data types and types of statements.
- Multiple / nested Mathematical and String functions.
- Nested Loops.
- Static/ non-static variables.
- Conversion of algebraic expressions to Java expressions.
- Syntax of switch-case and array declaration and initialization.

### **Suggestions for students**

- Computer Applications is a logic based subject like Mathematics and highly scoring in nature. It needs to be given time of self-study revision and practice at least 3 times a week for a duration of ½ an hour each day followed by a complete review of week's work during the week end. Do not leave this subject to be studied on the eve of the examination.
- All exercises, general and Programming to be maintained in a register and revised during exam preparation.
- Syntax of each element of Java to be thoroughly studied.
- At the time of answering, every question must be read at least twice before answering.
- All programs must be written with suitable comments – whether in class or during exam.
- A good variable description containing a list of important variables used in the program along with their data types and purpose must be given at the end of every program.