

COMPUTER APPLICATIONS

Answer Keys to Self-Assessment Sample Paper - 1

ANSWER KEY 1

- a) Encapsulation, Abstraction, Polymorphism, and Inheritance.
- b) Any two operators.
- c) The similarities are that each has a body of code that is executed perhaps multiple times. This is usually called the “loop body”. They also have a “while” portion that tests a condition that determines whether the loop should be executed another time. the differences between while and do while loop, ... Hence while loop executes the code block only if the condition is true. In do while loop, condition is tested at the end of the loop. So, Do While executes the statements in the code block at least once even if the condition fails.
- d) Statements are connected by more than one operator, like, and, or, not etc then it is called a compound statement. If a table contains more than one statement and representing true **value**, then it is called **truth** table. In compound statement, there are different rule for Conjunction, Disjunction and Negation.
- e) $a = 5$

ANSWER KEY 2

a) Default clause in switch statement works as the same as else clause in if else block. If any of the other case values of the switch statement is not true, the default case will be activated. A switch is a multiple-way statement that gives a simple way to send execution to various parts of codes based on the expression values.

If anything is not matched in a switch statement, then nothing could be done without the value default case. In a switch case, it is the exact significance of the default case.

b) An error in a program is called a bug and the process of removing it is called debugging.

c) equals - Compares this string to the specified object. compareTo compares two strings by their characters (at same index) and returns an integer (positive or negative) accordingly. equals() can be more efficient then compareTo(). equals() checks if two objects are the same or not and returns a boolean.

d) $z = 5 * x * x * x + 2 * y * x + y$

e)

```
import java.util.Scanner;
public class Alphabet {
```

```

public static void main(String[] args) {
    Scanner input = new scanner(System.in);
    char var ;
    System.outp.print("Enter A Charecter: ");
    var = input.nextChar();
    if( (var >= 'a' && var <= 'z') || (var >= 'A' && var <= 'Z'))
        System.out.println(var + " is an alphabet.");
    else
        System.out.println(var + " is not an alphabet.");
    }
}

```

ANSWER KEY 3

a) The **new operator** is used in Java to create **new** objects. **Keyword 'THIS'** in Java is a reference variable that refers to the current object. "this" is a reference to the current object, whose method is being called upon.

b) $x[5] \% 2 == 0$

c) A constructor in Java is a block of code like a method that is called when an instance of an object is created. A constructor is needed to initialise data members with legal initial values.

d) 1
7

e)

1-int n

2-1

3-i

4-n++

5- p=getresult(5)

f) 08

g) java. lang package
java.util

h) A string buffer is like a String but can be modified. It contains some particular sequence of characters, but the length and content of the sequence can be changed through certain method calls. They are safe for use by multiple threads. Every string buffer has a capacity.

ANSWER KEY 4

```
import java.util. * ;
```

```
class Telephone {
```

```
int prv,
pre,
call;
String name;
double amt,
total;
void input() {
    Scanner sc = new Scanner(System. in );
    System.out.println("Enter the previous meter reading: ");
    prv = sc.nextInt();
    System.out.println("Enter the present meter reading: ");
    pre = sc.nextInt();
    System.out.println("Enter the name: ");
    name = sc.nextLine();
}
void cal() {
    call = pre - prv;
    if (call <= 100) amt = 0;
    else if (call > 100 && call <= 200) amt = 0 * 100 + (call - 100) * 0.90;
    else if (call > 200 && call <= 400) amt = 0 * 100 + 100 * 0.90 + (call - 200) * 0.80;
    else amt = 0 * 100 + 100 * 0.90 + 200 * 0.80 + (call - 400) * 0.70;
    total = amt + 180;
}
void display() {
    System.out.println("Name\t\tCalls Made\t\tAmount\t\tTotal Amount");
```

```
System.out.println(name + "\t\t" + call + "\t\t" + amt + "\t\t" + total);
}

public static void main(String args[] ) {

    Telephone ob = new Telephone();

    ob.input();

    ob.cal();

    ob.display();

}

}
```

ANSWER KEY 5

```
import java.io. * ;
class StringSort {
    public static void main(String a[]) throws IOException {
        BufferedReader br = new BufferedReader(new InputStreamReader(System. in ));
        System.out.print("Enter a String: ");
        String ao = br.readLine();
        char[] arr = ao.toCharArray();
        int l = ao.length();
        char temp;
        for (int i = 0; i < l - 1; i++) {
            for (int j = 0; j < l - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
        String a1 = String.valueOf(arr);
        System.out.print("String after Sorting: " + a1);
    }
}
```

ANSWER KEY 6

```
import java.util.Scanner;

public class BubbleSort {
```

```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    System.out.println("Enter ten numbers:");
    int[] numbers = new int[10];
    for (int i = 0; i < 10; i++) {
        numbers[i] = scanner.nextInt();
    }
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10 - i - 1; j++) {
            if (numbers[j] < numbers[j + 1]) {
                int temp = numbers[j];
                numbers[j] = numbers[j + 1];
                numbers[j + 1] = temp;
            }
        }
    }
    System.out.println("Sorted Numbers:");
    for (int i = 0; i < 10; i++) {
        System.out.println(numbers[i]);
    }
}
}

```

ANSWER KEY 7

```
import java.util.Scanner;
```

```

public class Menu {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("1. Floyd's triangle");
        System.out.println("2. ICSE Pattern");
        System.out.print("Enter your choice: ");
        int choice = scanner.nextInt();
        switch (choice) {
            case 1:
                System.out.print("Enter n (number of lines): ");
                int n = scanner.nextInt();
                int currentNumber = 1;
                for (int i = 1; i <= n; i++) {
                    for (int j = 1; j <= i; j++) {
                        System.out.print(currentNumber + " ");
                        currentNumber++;
                    }
                    System.out.println();
                }
                break;
            case 2:
                System.out.print("Enter word: ");
                String word = scanner.next();

```

```

        for (int i = 0; i < word.length(); i++) {
            for (int j = 0; j <= i; j++) {
                System.out.print(word.charAt(j) + " ");
            }
            System.out.println();
        }
        break;
    default:
        System.out.println("Invalid choice");
        break;
    }
}
}

```

ANSWER KEY 8

```

import java.io.*;
class Letters{
    public static void main(String args[])throws IOException{
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Enter the sentence: ");
        String s = br.readLine().toUpperCase();
        int count = 0;
        String word = new String();
        for(int i = 0; i < s.length(); i++){
            char ch = s.charAt(i);
            switch(ch){
                case ' ':
                case '.':
                case '?':
                case '!':
                    if(word.charAt(0) == 'A')
                        count++;
                    word = new String();
                    break;
                default:
                    word += ch;
            }
        }
        System.out.println("Total number of words starting with letter 'A' = " + count);
    }
}

```

ANSWER KEY 9

```

class coprime {

    static void main(int n, int m) {

```

```
int k = Math.min(n, m);

int v = 0;

for (int a = 1; a <= k; a++) {

    if (n % a == 0 && m % a == 0) {

        v++;

    }

}

if (v == 1) {

    System.out.println("co-prime number");

}

else {

    System.out.println("Not a co-prime number");

}

}
```

COMPUTER APPLICATIONS

Answer Keys to Self-Assessment Sample Paper - 2

ANSWER KEY 1

(a) abstraction refers to the act of representing essential features without including the background details or explanations.

(b) Searching allows us to find an element in an array.
Sorting allows us to arrange the array elements in a particular order.

(c) The `isUpperCase()` method checks if a given character is in uppercase or not.
The `toUpperCase()` method converts a given character to uppercase.

(d) The public members can be accessed by anyone.
The private members allow only the methods of the same class to access them.

(e)
(i) char – primitive data type.
(ii) arrays – non-primitive data type.
(iii) int – primitive data type.
(iv) classes – non-primitive data type.

ANSWER KEY 2

(a) ++, %, >=, &&

(b)
(i) `System.out.println("Java");` Method invocation
(ii) `costPrice = 457.50;` Assignment
(iii) `Car hybrid = new Car();` Object creation
(iv) `petrolPrice++;` Increment

(c)
(i) The switch statement tests for one variable, whereas the if-else statement tests for multiple variables.
(ii) The switch statement only tests for equality, whereas the if-else statement can test for any type of condition.

(d) An infinite loop is one that keeps on executing repeatedly because its terminating condition is either missing or is never reached.

```
for(;;)  
    System.out.println("Java");
```

(e) A constructor is a member function with the same name as its class. It is used to initialize the objects of the class with legal initial values.
It is invoked at the time of object creation.

ANSWER KEY 3**(a)**

(i) "MISSISSIPPI".indexOf('S') + "MISSISSIPPI".lastIndexOf('I')

2 + 10 = 12

(ii) "CABLE".compareTo("CADET")

66 - 68 = -2

(b)

(i) Math.ceil(4.2)

5.0

(ii) Math.abs(-4)

4

(c) Parameterized constructor are the ones that receive parameters and initialize objects with received values.**(d)** $T = \text{Math.sqrt}(A * A + B * B + C * C)$ **(e)**

System.out.print(((x % 2 == 0)? "EVEN":"ODD"));

(f)

int m = 5, n = 10;

for(; n >= 1; n--)

System.out.println(m * n);

(g) Primitive data types come as part of the language.

Composite data types are constructed from primitive data types.

(h)

OUTPUT:

5

10

The body of the loop executes 3 times.

(i) a += a++ + ++a + --a + a--; when a = 7.

a = a + a++ + ++a + --a + a--;

= 7 + 7 + 9 + 8 + 8

= 14 + 9 + 8 + 8

= 23 + 8 + 8

= 31 + 8

= 39.

(j) (i) isLetterOrDigit(char) – boolean

(ii) replace(char, char) – String

ANSWER KEY 4

import java.io.*;

class BookFair{

```
String bName;
double price;
public BookFair(){
    bName = new String();
    price = 0.0;
}
public void input()throws IOException{
    InputStreamReader in = new InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(in);
    System.out.print("Book name: ");
    bName = br.readLine();
    System.out.print("Book price: ");
    price = Double.parseDouble(br.readLine());
}
public void calculate(){
    if(price <= 1000.0) price -= 2.0 / 100 * price; else if(price > 1000.0 && price <= 3000.0)
        price -= 10.0 / 100 * price;
    else
        price -= 15.0 / 100 * price;
}
public void display(){
    System.out.println("Book name: " + bName);
    System.out.println("Book price: Rs. " + price);
}
public static void main(String args[])
throws IOException{
    BookFair obj = new BookFair();
    obj.input();
    obj.calculate();
    obj.display();
}
}
```

ANSWER KEY 5

```
import java.util.Scanner;

public class KboatExpression
{
    public static void main(String args[] ) {

        Scanner in = new Scanner(System.in);

        System.out.print("Enter the value of a: ");
        int a = in.nextInt();

        System.out.print("Enter the value of b: ");
        int b = in.nextInt();

        System.out.print("Enter the value of c: ");
```

```
int c = in.nextInt();

double exp = (1 / Math.pow(a, 2)) + (2 / Math.pow(b, 2)) + (3 / Math.pow(c, 2));
long roundedExp = Math.round(exp);

System.out.println("Result rounded to whole number: " + roundedExp);
}
}
```

ANSWER KEY 6

```
import java.util.Scanner;

public class Kboat20LetterSet
{
    public void findVowelNConsonants() {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter any 20 letters");
        int vc = 0, cc = 0;
        for (int i = 0; i < 20; i++) {
            char ch = in.next().charAt(0);
            ch = Character.toUpperCase(ch);
            if (ch == 'A' ||
                ch == 'E' ||
                ch == 'I' ||
                ch == 'O' ||
                ch == 'U')
                vc++;
            else if (ch >= 'A' && ch <= 'Z')
                cc++;
        }
        System.out.println("Number of Vowels = " + vc);
        System.out.println("Number of Consonants = " + cc);
    }
}
```

ANSWER KEY 7

```
import java.util.Scanner;

public class KboatSurnameFirst
{
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a name of 3 words:");
        String name = in.nextLine();

        /*
         * Get the last index
         * of space in the string
         */
    }
}
```

```

int lastSpaceIdx = name.lastIndexOf(' ');

String surname = name.substring(lastSpaceIdx + 1);
String initialName = name.substring(0, lastSpaceIdx);

System.out.println(surname + " " + initialName);
}
}

```

ANSWER KEY 8

```

class Overload{
    public static double volume(double r){
        return 4.0 / 3 * 22.0 / 7 * r * r * r;
    }
    public static double volume(double h, double r){
        return 22.0 / 7 * r * r * r * h;
    }
    public static double volume(double l, double b, double h){
        return l * b * h;
    }
}

```

ANSWER KEY 9

```

import java.io.*;
class Menu{
    public static void main(String args[])
    throws IOException{
        InputStreamReader in = new InputStreamReader(System.in);
        BufferedReader br = new BufferedReader(in);
        System.out.println("1. Pattern 1");
        System.out.println("2. Pattern 2");
        System.out.print("Enter your choice: ");
        int choice = Integer.parseInt(br.readLine());
        switch(choice){
            case 1:
                String s = "ABCDE";
                for(int i = s.length(); i >= 0; i--){
                    System.out.println(s.substring(0, i));
                }
                break;
            case 2:
                s = "BLUE";
                for(int i = 0; i < s.length(); i++){
                    char ch = s.charAt(i);
                    for(int j = 0; j <= i; j++){
                        System.out.print(ch);
                    }
                    System.out.println();
                }
                break;
            default:

```

```
        System.out.println("Invalid choice!");  
    }  
}  
}
```

COMPUTER APPLICATIONS

Answer Keys to Self-Assessment Sample Paper - 3

ANSWER KEY 1

a) Data Hiding and Data Abstraction are complementary concepts. Data Abstraction focuses on the observable behaviour of an object, whereas Data hiding, or Data Encapsulation focuses upon the implementation that gives rise to this behaviour. In other words, Data Abstraction cares about what something does but not how it does it. Data Encapsulation cares about how something does what it does such that others do not have to worry about the implementation details. Hence, we can say that Encapsulation is a way to implement Data Abstraction.

b) This statement creates a new object of class Employee. The newly created object is assigned to a variable named staff which is of Employee type. The object can be accessed using staff variable.

c) A token is the smallest element of a program that is meaningful to the compiler. The different types of tokens in Java are:

Identifiers

Literals

Operators

Separators

Keywords

d)

Unary Arithmetic Operator	Binary Arithmetic Operator
It operates on a single operand	It operates on two operands
Increment (++) and Decrement (--) operators are examples of Unary Arithmetic Operators	Multiplication (*) and Division (/) are examples of Binary Arithmetic Operators

e) Java Virtual Machine takes Bytecode as input and converts it into Machine Code one line at a time. This Bytecode can be generated by compiling source code written in any JVM language like Scala, Kotlin, etc not just Java. Hence, Java interpreter is called Java Virtual Machine.

ANSWER KEY 2

a) Java provides the following ways to give input in a program:

Using Function Argument.

Using InputStreamReader class.

Using Scanner class.

Using Command Line Arguments.

b) -5.0

-6.0

c) `System.exit(0)` terminates the execution of the program by stopping the Java Virtual Machine which is executing the program. It is generally used when due to some reason it is not possible to continue with the execution of the program.

d) for loop is an entry-controlled loop. The following parameters are commonly used in a for loop:
An initial value for the loop control variable.

A condition—loop will iterate as long as this condition remains true.

An update expression to modify the loop control variable after every iteration.

Body of the loop which consists of the statements that needs to be repeatedly executed.

e) 'break outer' will terminate the loop that is labelled as outer in a nested loop and transfer the program control to the statement just after the loop labelled as outer.

'continue outer' will skip the remaining statements of the nested loop and start the next iteration of the loop that is labelled as outer.

ANSWER KEY 3

a) `Float.parseFloat()`

It is used to convert string data into float data.

`Double.toString()`

It is used to convert double data to a string.

b) The data type that represents a number of similar or different data under single declaration is called as composite data type. An array is a group or a collection of same type of variables. Hence, Array is a composite data type.

c)

(i) Extract the second last character of a word stored in the variable `wd`.

```
char ch = wd.charAt(wd.length() - 2);
```

(ii) Check if the second character of a string `str` is in upper case.

```
boolean res = Character.isUpperCase(str.charAt(1));
```

d)

Two ways of invoking functions are:

Pass by value.

Pass by reference.

e) When a class is declared with public access specifier it is said that the class is publicly accessible. A publicly accessible class is visible everywhere both within and outside its package. For example:

```
public class Example {  
    //Class definition  
}
```

f)

(a) Constructor with default argument

Java specification doesn't support default arguments in methods so Constructor with default argument cannot be written in Java.

(b) Parameterised constructor

A Parameterised constructor receives parameters at the time of creating an object and initializes the object with the received values.

(c) Copy constructor

A constructor used to initialize the instance variables of an object by copying the initial values of the instance variables from another object is known as Copy Constructor.

(d) Constructor overloading

The process of using a number of constructors with the same name but different types of parameters is known as Constructor overloading.

g)

Output

x = 2

y = 2

h)

Below is the syntax of nested loop:

```
for (<initial value>; <test condition>; <update value>) {  
    for (<initial value>; <test condition>; <update value>) {  
        executable statement(s)  
    }  
}
```

i) A class member declared as protected can be accessed directly by all the classes within the same package and the subclasses of its class even if the subclasses are in different packages.

Example:

```
class A {  
    protected int amount;  
}  
class B extends A {  
    public void displayAmount() {  
        System.out.println("Amount = " + amount);  
    }  
}
```

ANSWER KEY 4

```
import java.util.Scanner;
```

```
public class Factorial  
{  
    private int n;
```



```
public void input() {
    Scanner in = new Scanner(System.in);
    System.out.print("Enter the number: ");
    n = in.nextInt();
}
```

```
public void fact() {
    int f = 1;
    for (int i = 1; i <= n; i++)
        f *= i;
    System.out.println("Factorial of " + n
        + " = " + f);
}
```

```
public static void main(String args[]) {
    Factorial obj = new Factorial();
    obj.input();
    obj.fact();
}
}
```

ANSWER KEY 5

```
import java.util.Scanner;
```

```
public class Laptop
{
    private String name;
    private int price;
    private double dis;
    private double amt;

    public Laptop(String s, int p)
    {
        name = s;
        price = p;
    }
}
```

```
public void compute() {
    if (price <= 25000)
        dis = price * 0.05;
    else if (price <= 50000)
        dis = price * 0.075;
    else if (price <= 100000)
        dis = price * 0.1;
    else
        dis = price * 0.15;
}
```

```
    amt = price - dis;
}

public void display() {
    System.out.println("Name: " + name);
    System.out.println("Discount: " + dis);
    System.out.println("Amount to be paid: " + amt);
}

public static void main(String args[]) {

    Scanner in = new Scanner(System.in);
    System.out.print("Enter Customer Name: ");
    String str = in.nextLine();
    System.out.print("Enter Price: ");
    int p = in.nextInt();

    Laptop obj = new Laptop(str,p);
    obj.compute();
    obj.display();
}
}
```

ANSWER KEY 6

```
import java.util.Scanner;

public class KboatArmstrongNumber
{
    public int armstrong(int n) {

        int num = n, cubeSum = 0;

        while (num > 0) {
            int digit = num % 10;
            cubeSum = cubeSum + (digit * digit * digit);
            num /= 10;
        }

        if (cubeSum == n)
            return 1;
        else
            return 0;
    }

    public static void main(String args[]) {

        Scanner in = new Scanner(System.in);
        System.out.print("Enter Number: ");
```

```
int num = in.nextInt();

KboatArmstrongNumber obj = new KboatArmstrongNumber();
int r = obj.armstrong(num);

if (r == 1)
    System.out.println(num + " is an Armstrong number");
else
    System.out.println(num + " is not an Armstrong number");
}
}
```

ANSWER KEY 7

```
import java.util.Scanner;

public class KboatWordsNLetters
{
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a sentence:");
        String str = in.nextLine();

        int wCount = 0, lCount = 0;
        int len = str.length();
        for (int i = 0; i < len; i++) {
            char ch = str.charAt(i);
            if (ch == ' ')
                wCount++;
            else
                lCount++;
        }

        /*
        * Number of words in a sentence are one more than
        * the number of spaces so incrementing wCount by 1
        */
        wCount++;

        System.out.println("No. of words = " + wCount);
        System.out.println("No. of letters = " + lCount);
    }
}
```

ANSWER KEY 8

```
import java.util.Scanner;

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

```
Scanner in = new Scanner(System.in);

System.out.println("Please enter the 3 sides of the triangle.");

System.out.print("Enter the first side: ");
double a = in.nextDouble();

System.out.print("Enter the second side: ");
double b = in.nextDouble();

System.out.print("Enter the third side: ");
double c = in.nextDouble();

double p = a + b + c;
double s = p / 2;
double area = Math.sqrt(s * (s - a) * (s - b) * (s - c));

if (area == p)
    System.out.println("Entered triangle is equable.");
else
    System.out.println("Entered triangle is not equable.");
}
```

ANSWER KEY 9

```
import java.util.Scanner;

public class KboatSalary
{
    public static void main(String args[]) {

        Scanner in = new Scanner(System.in);

        System.out.print("Enter the Salary: ");
        float salary = in.nextFloat();

        float foodSpend = salary / 2;
        float rentSpend = salary / 15;
        float miscSpend = salary / 10;
        float savings = salary - (foodSpend + rentSpend + miscSpend);

        System.out.println("Money spent on food: " + foodSpend);
        System.out.println("Money spent on rent: " + rentSpend);
        System.out.println("Money spent on miscellaneous: " + miscSpend);
        System.out.println("Money saved: " + savings);
    }
}
```

COMPUTER APPLICATIONS

Answer Keys to Self-Assessment Sample Paper - 4

ANSWER KEY 1

a) Accumulator is a variable that is used to add or accumulate a list of items. Counter on the other hand is a variable, which is used to keep track of the number of times an operation is being performed.

b) $y=33$

c) A comment is a programmer-readable explanation or annotation in the source code of a computer program.

Types: Single-line comment and Multiline comment

d). Data Abstraction is the property by virtue of which only the essential details are displayed to the user.

For example, when we ride a bike, we concentrate only driving rather than how the engine is working according to the different switches, lever, and steering.

e) Function prototype is the first line of the function definition that consist of return type function name and the parameter list. Function signature on the other hand only specifies the parameter list.

ANSWER KEY 2

a) `switch(var)`

```
{
case 1:
System.out.println("good");
break;
case 2:
System.out.println("better");
break;
case 3:
System.out.println("best");
break;
default:
System.out.println("invalid");
}
```

b) $m=3$

$n=14$

c) $s=u*t+1/2.0*a*Math.pow(t,2)$

d) Following are the difference between constructor and method.

a. Constructor is used to initialize an object whereas method is used to exhibits functionality of an object.

- b. Constructors are invoked implicitly whereas methods are invoked explicitly.
- c. Constructor does not return any value where the method may/may not return a value.
- d. In case constructor is not present, a default constructor is provided by java compiler. In the case of a method, no default method is provided.
- e. Constructor should be of the same name as that of class. Method name should not be of the same name as that of class.

e) Example of how objects are passed as reference:

```
class PassByReference
{
static void change(Number r)
{
r.a=r.a+5;
r.b=r.b+5;
}
static void call()
{
Number n=new Number(5,6);
System.out.println(n.a+"\t"+n.b);
change(n);
System.out.println(n.a+"\t"+n.b);
}
}
```

will result in the following output when the call() function is executed:

```
5 6
10 1
```

ANSWER KEY 3

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

- (i) static int x;
- (ii) arr[i] = 10; (Composite)
- (iii) obj.display(); (Composite)
- (iv) boolean b;
- (v) private char chr;
- (vi) String str; (Composite)

(b)

```
grinds
WHEAT
```

(c)

```
x = 6.0
y = 15.0
```

(d)

```
String grade = "";
if(marks >= 90)
    grade = "A";
else if(marks >= 80)
```

```

    grade = "B";
else
    grade = "C";

```

(e)

6

4

(f)

(i) compareTo()

int

(ii) equals()

Boolean

(g)

characteristic = 4

manissa = 3756

(h)

(i) How many times does the loop execute?

6 times.

(ii) What is the range of possible values stored in the variable number?

The range is 0 to 9.

(i)

(i) Name the variables for which each object of the class will have its own distinct copy.

a and b.

(ii) Name the variables that are common to all objects of the class.

x and y.

(j)

x = 1001

y = 1001.0

The King said "Begin at the beginning!" to me.

ANSWER KEY 4

```

import java.util. * ;
class Sol10 {
    static void main() {
        Scanner sc = new Scanner(System. in );
        float tc,
        d,
        ap;
        System.out.println("Enter the total cost of the items: ");
        tc = sc.nextFloat();
        if (tc <= 2000) d = 5 / 100f * tc;
        else if (tc >= 2001 && tc <= 5000) d = 25 / 100f * tc;
        else if (tc >= 5001 && tc <= 10000) 39 Revision Tour II d = 35 / 100f * tc;
    }
}

```

```

else d = 50 / 100f * tc;
ap = tc - d;
System.out.println("Amount Payable: " + ap);
}
}

```

ANSWER KEY 5

```

import java.util. * ;
class Sol37 {
    static void main() {
        Scanner sc = new Scanner(System. in );
        int i,
        j,
        f = 0,
        n;
        System.out.println("Enter a number: ");
        n = sc.nextInt();
        for (i = n; i > 0; i /= 10)
            f++;
        if ((n * n) % (int) Math.pow(10, f) == n) System.out.println("Automorphic Number");
        else if ((n * n * n) % (int) Math.pow(10, f) == n) System.out.println("Trimorphic Number");
        else if ((3 * n * n) % (int) Math.pow(10, f) == n) System.out.println("Tri - automorphic
Number");
        else System.out.println("Not Automorphic, Trimorphic or Tri - automorphic Number");
    }
}

```

ANSWER KEY 6

```

import java.util. * ;
class Q6 {
    static void main() {
        Scanner sc = new Scanner(System. in );
        int i,
        s = 0,
        ch,
        x;
        float sum = 0;
        System.out.println("M E N U");
        System.out.println("1.First Series");
        System.out.println("2.Second Series");
        System.out.println("Enter your choice: ");
        ch = sc.nextInt();
        switch (ch) {
            case 1:
                for (i = 2; i <= 18; i += 4)
                    s += i - (i + 2);
                System.out.println("Sum = " + s);
                break;

```



```

case 2:
    System.out.println("Enter the value of x:
    ");
    x = sc.nextInt();
    for (i = 2; i <= 20; i += 3)
        sum += (float) x / i;
    System.out.println("Sum: " + sum);
    break;
default:
    System.out.println("Invalid choice ! ");
}
}
}

```

ANSWER KEY 7

```

import java.util. * ;
class Mobike {
    String bno,
    phno,
    name;
    int days;
    double charge;
    void input() {
        Scanner sc = new Scanner(System. in );
        System.out.println("Enter your bike no.: ");
        bno = sc.nextLine();
        System.out.println("Enter your phone no.: ");
        phno = sc.nextLine();
        System.out.println("Enter your name: ");
        name = sc.nextLine();
        System.out.println("Enter no.of days taken
        for rent: ");
        days = sc.nextInt();
    }
    void display() {
        System.out.println("Bike No.\t\tPhone No.\t\tName\t\tNo.of days\t\tCharge");
        System.out.println(bno + "\t\t" + phno + "\t\t" + name + "\t\t" + days + "\t\t" + charge);
    }
    void calc() {
        if (days <= 5) charge = days * 500;
        else if (days > 5 && days <= 10) charge = 5 * 500 + (days - 5) * 400;
        else charge = 5 * 500 + 5 * 400 + (days - 10) * 200;
    }
}
}

```

ANSWER KEY 8

```
import java.util. * ;
class Sol3 {
    static boolean isPerfect(int n) {
        int i,
        s = 0;
        for (i = 1; i < n; i++) {
            if (n % i == 0) s += i;
        }
        if (s == n) return true;
        else return false;
    }
    static void main() {
        Scanner sc = new Scanner(System. in );
        int a,
        b;
        System.out.println("Enter 2 numbers: ");
        a = sc.nextInt();
        b = sc.nextInt();
        if (isPerfect(a) == true && isPerfect(b) == true) System.out.println("Both are prefect numbers");
        else System.out.println("Both are not perfect numbers");
    }
}
```

ANSWER KEY 9

```
class FourSide {
    int length,
    breadth;
    FourSide(int s) {
        length = breadth = s;
    }
    FourSide(int l, int b) {
        length = l;
        breadth = b;
    }
    void compute() {
        int area;
        area = length * breadth;
        System.out.println("Area: " + area);
    }
    public static void main(String args[]) {
        FourSide ob1 = new FourSide(12);
        FourSide ob2 = new FourSide(17, 6);
        ob1.compute();
        ob2.compute();
    }
}
```

COMPUTER APPLICATIONS

Answer Keys to Self-Assessment Sample Paper - 5

ANSWER KEY 1

a) A Wrapper class is a class whose object wraps or contains a primitive data type. Wrapper classes used in Java- Byte, Short, Integer, Long, Character, Boolean, Float and Double.

b) In **pure function** it does not modify the external variable/data outside the scope and results the same output given in the same input. In **impure function** it mutates data/variable outside its lexical scope.

In pure expressions, all the operands are of same type. And in mixed expressions, the operands are of mixed or different data types.

c) Type casting is nothing but assigning a value of one [primitive data type](#) to another. When you assign the value of one data type to another, you should be aware of the compatibility of the data type. If they are compatible, then [Java](#) will perform the conversion automatically known as Automatic Type Conversion and if not, then they need to be casted or converted explicitly.

d) A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

e) m=6, n=14

ANSWER KEY 2

a) helloworld

5

1

false

b) The **nextDouble()** method of [java.util.Scanner](#) class scans the next token of the input as a Double. If the translation is successful, the scanner advances past the input that matched. The **hasNextDouble()** method of [java.util.Scanner](#) class returns true if the next token in this scanner's input can be interpreted as a Double using the nextDouble() method. The scanner does not advance past any input.

c) import keyword is used to import built-in and user-defined packages into our Java program.

d) Omitting break statement after a case leads to program execution continuing into the next case and onwards till a break statement is encountered or end of switch is reached. To avoid this, we must use break after each case in a switch statement.

ANSWER KEY 3

a) == operator is used to compare two or more than two objects, If they are referring to the same object then return true, otherwise return false. String is immutable in java. When two or more objects are created without new keyword, then both object refer same value. Double equals operator actually compares objects references.

Java String equalsIgnoreCase() method is much similar to equals() method, except that case is ignored like in above example String object s4 compare to s3 then equals() method return false, but here in case of equalsIgnoreCase() it will return true. Hence equalsIgnoreCase() method is Case Insensitive.\

b) To create a package:

Step 1: Select the Edit menu and select the New Package... option.

Step 2: Enter the name of the package in the 'Create New Package' dialog and hit the 'OK' button.

c) $(\text{Math.sqrt}(3) * a * a) / 4$

$\text{Math.cbrt}(m * n) + (m * \text{Math.pow}(n, 3))$

d) In Call by Value method value of actual parameter are copied into the formal parameter that is the function creates its own copy of argument values and works on this value whatever change takes place are not reflected back to the original own actual variable or parameter because function does not have access to the original variables this type of calling is useful if original values are not to be modified.

e) 11

12

13

14

15

16

f) 0

0

-2

-6

-12

Loop executes 5 times

g) `grade=(avg>= 40 && avg<=100) ?'P' : 'F';`

h)

No.	throw	throws
1)	Java throw keyword is used to explicitly throw an exception.	Java throws keyword is used to declare an exception.
2)	Checked exception cannot be propagated using throw only.	Checked exception can be propagated with throws.

3)	Throw is followed by an instance.	Throws is followed by class.
4)	Throw is used within the method.	Throws is used with the method signature.
5)	You cannot throw multiple exceptions.	You can declare multiple exceptions e.g. public void method()throws IOException,SQLException.

```
i) Boolean chk(int x, int y){
return x>y ? True : false;
}
```

ANSWER KEY 4

```
#include < iostream > #include < string > using namespace std;

class student {

char s_name[50];

int s_rno;

float s_marks[5];

char s_stream[25];

void assign() {

float average = this ->avg();

if (average > 70 && average < 76) strcpy(stream, "Civil", 25);

else if (average > 75 && average < 81) strcpy(stream, "Chemical", 25);

else if (average > 80 && average < 86) strcpy(stream, "Electrical", 25);

else if (average > 85 && average < 91) strcpy(stream, "Mechanical", 25);

else if (average > 90 && average < 96) strcpy(stream, "Electronics", 25);

else if (average >= 96) strcpy(stream, "Computer Science", 25);

else strcpy(stream, "No stream", 25);

}
```

```
public: student() {};  
  
student(student & r) {  
  
    rno = r.rno;  
  
    for (int i = 0; i < 5; i++, marks[i] = r.marks[i]);  
  
    strncpy(stream, r.stream, 25);  
  
    strncpy(name, r.name, 50);  
  
}  
  
void input() {  
  
    cout << "Enter roll no\n";  
  
    cin >> rno;  
  
    cin.ignore(1, '\n');  
  
    cout << "ENter name\n";  
  
    cin.getline(name, 50);  
  
    cout << "Enter marks in five subjects\n";  
    for (int i = 0; i < 5; i++, cin >> marks[i]);  
  
    assign();  
  
}  
  
void display() {  
  
    cout << "\nROll no: " << rno << endl;  
  
    cout << "Name: " << name << endl;  
  
    cout << "Average : " << student::avg() << " %" << endl;  
  
    cout << "Stream: " << stream << endl;  
  
}  
  
float avg() {  
  
    float average = 0;
```

```
for (int i = 0; i < 5; i++, average += marks[i]);

average /= 5;

cout << "FUNCTION AVERAGE(): AVG= " << average << endl; //Diagnostic message

return (average);

}

void search(student * , int n);

};

void student::search(student * a, int n) {

    int i = 0;

    int found = 0;

    char nm[50];

    cin.ignore(1, '\n');

    cout << "ENter name to search\n";

    cin.getline(nm, 50);

    while (i < n && (!found)) {

        if (strcmp(nm, a[i].name) == 0) {

            found = 1;

        }

        i++;

    }

    if (found) a[i].display();

    else cout << "NOT FOUND\n\n";

}

void highavg(student * stud, int n) {

    cout << n << endl;
```

```
float max = stud[0].avg();
int p = 0;
for (int i = 1; i < n; i++) {
    if (max < stud[i].avg()) {
        max = stud[i].avg();
        p = i;
    }
}
```

ANSWER KEY 5

```
import java.util. * ;
class dr {
    static void main(long n) {
        Scanner scr = new Scanner(System. in );
        System.out.println("enter");
        String s[] = new String[25];
        long a[] = new long[25];
        int i;
        for (i = 0; i <= 24; i++) {
            s[i] = scr.nextLine();
        }
        int high = 24;
        int low = 0;
        while (low < high) {
            int mid = (low + high) / 2;
```



```
if (n == a[mid]) {  
    System.out.println(a[mid] + s[mid]);  
} else if (n > a[mid]) {  
    low = mid + 1;  
} else if (n < a[mid]) {  
    high = mid - 1;  
}  
}  
}  
}
```

ANSWER KEY 6

```
import java.util. * ;  
class Q6 {  
    static void isComposite(int n) {  
        int i,  
        c = 0;  
        for (i = 1; i <= n; i++) {  
            if (n % i == 0) c++;  
        }  
        if (c > 2) System.out.println("Composite number");  
        else System.out.println("Not a Composite number");  
    }  
    static void smallest(int n) {  
        int i,  
        d,  
        min = 0;  
        for (i = n; i > 0; i = i / 10) {  
            d = i % 10;  
            if (min == 0) min = d;  
            if (d < min) min = d;  
        }  
    }  
}
```

```
System.out.println("Smallest digit: " + min);
}
static void main() {
    Scanner sc = new Scanner(System. in );
    int i,
    n,
    r,
    ch,
    d;
    System.out.println("M E N U");
    System.out.println("1.Composite Number");
    System.out.println("2.Smallest digit in a number");
    System.out.println("Enter your choice: ");
    ch = sc.nextInt();
    switch (ch) {
    case 1:
        System.out.println("Enter a number:
        ");
        n = sc.nextInt();
        isComposite(n);
        break;
    case 2:
        System.out.println("Enter a number:
        ");
        n = sc.nextInt();
        smallest(n);
        break;
    default:
        System.out.println("Invalid choice ! ");
    }
}
}
```

ANSWER KEY 7

```
import java.util.Scanner;

public class Q7
```

```
{
public static void main(String args[]) {
    Scanner in = new Scanner(System.in);
    System.out.println("Enter the number: ");
    int num = in.nextInt();
    int copyNum = num;
    int revNum = 0;

    while(copyNum != 0) {
        int digit = copyNum % 10;
        copyNum /= 10;
        revNum = revNum * 10 + digit;
    }

    if (revNum == num)
        System.out.println("A Palindrome number");
    else
        System.out.println("Not a Palindrome number");
}
}
```

ANSWER KEY 8

```
import java.util.Scanner;
```

```
public class KboatResistance
```

```
{
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        System.out.println("1. Series");
        System.out.println("2. Parallel");

        System.out.print("Enter your choice: ");
        int choice = in.nextInt();
        boolean isChoiceValid = true;

        System.out.print("Enter r1: ");
        double r1 = in.nextDouble();
        System.out.print("Enter r2: ");
        double r2 = in.nextDouble();
        double eqr = 0.0;

        switch (choice) {
            case 1:
                eqr = r1 + r2;
                break;
            case 2:
                eqr = (r1 * r2) / (r1 + r2);
                break;
            default:
```

```
        isChoiceValid = false;
        System.out.println("Incorrect choice");
        break;
    }

    if (isChoiceValid)
        System.out.println("Equivalent resistance = " + eqr);
}
}
```

ANSWER KEY 9

```
import java.util.Scanner;

public class KboatPattern
{
    public static void main(String args[]) {
        Scanner in = new Scanner(System.in);
        System.out.println("Type 1 for a triangle");
        System.out.println("Type 2 for an inverted triangle");

        System.out.print("Enter your choice: ");
        int ch = in.nextInt();

        switch (ch) {
            case 1:
                for (int i = 1; i <= 5; i++) {
                    for (int j = 1; j <= i; j++) {
                        System.out.print(i + " ");
                    }
                    System.out.println();
                }
                break;

            case 2:
                for (int i = 5; i > 0; i--) {
                    for (int j = 1; j <= i; j++) {
                        System.out.print(i + " ");
                    }
                    System.out.println();
                }
                break;

            default:
                System.out.println("Incorrect Choice");
                break;
        }
    }
}
```