University of Colombo, Sri Lanka

University of Colombo School of Computing

DEGREE OF BACHELOR OF INFORMATION TECHNOLOGY (EXTERNAL)

Academic Year 2024 — 1st Year Examination — Semester 1

IT1406 — Introduction to Programming

Multiple Choice Question Paper

(2 Hours)

Important Instructions

- The duration of the paper is **2 Hours**.
- The medium of instructions and questions is English.
- This paper has **40 questions** on **12 pages**. Answer **all** questions.
- All questions are of the **MCQ** (Multiple Choice Questions) type.
- Each question will have **5 (five)** choices with **ONLY ONE** correct answer.
- This paper consists of 100 marks and all the questions will carry equal marks.
- Answers should be marked on the **special answer sheet** provided.
- Note that questions appear on both sides of the paper. If a page or part of a page is not printed, please inform the supervisor/invigilator immediately.
- Mark the correct choices on the question paper first and then transfer them to the given answer sheet which will be machine marked. **Please completely read and follow the instructions given on the other side of the answer sheet before you shade your correct choices.**
- Any electronic device capable of storing and retrieving text, including electronic dictionaries, smartwatches, and mobile phones, is not allowed.
- Calculators are **not** allowed.
-

1). Which of the following statements is ***correct*** regarding the ***Java programming language***?

a). Java is a direct descendant of the C programming language.

b). The output of a Java compiler is executable code.

c). The Java Virtual Machine (JVM) serves as an interpreter for bytecode.

d). Java programs are compiled and interpreted at the same time during runtime.

e). The Java Virtual Machine (JVM) serves as a compiler for bytecode and executable code.

2). Which of the following is a ***correct*** example of a Java identifier?

a). 2Sum b). -Sum c). _Sum

d). Sum\ e). S/um

3). Which of the following statements is correct regarding ***primitive*** data types of Java?

a). There are automatic conversions from numeric to Boolean types.

b). Primitive data types can be used to represent complex objects.

c). There are three primitive data types that store integers.

d). The values of *true* and *false* cannot be converted into a numerical representation.

e). The Java compiler checks all expressions and parameters to ensure that the types are compatible at runtime.

4). Which of the following statements is ***correct*** regarding ***module cohesion***?

a). Cohesion is a measure of the number of methods in a module.

b). It indicates how different the elements of a module are from each other.

c). The more the elements of a module are independent, the lower the cohesion of the module.

d). Modules with high cohesion are considered conflicting modules because of their different responsibilities.

e). High cohesion in a module leads to increased dependency on other modules.

5). Which of the following statements is ***correct*** regarding the given Java array??

```
int twoDMatrix[ ][ ]={
{5,1,2,8,6},{8,1,6,2},{3,9,9,5,4},{1,7,3,6,4}};
```

a). System.out.println(twoDMatrix.length); outputs 5.
b). System.out.println(twoDMatrix[2].length); outputs 3.
c). System.out.println(twoDMatrix[3][2]); outputs 3.
d). System.out.println(twoDMatrix[0][6]); outputs 9.
e). System.out.println(twoDMatrix[2].length); outputs 4.

6). Which of the following statements is ***correct*** regarding Java ***Abstract Window Toolkit (AWT)*** ?

(I) It is used to create graphical user interfaces in Java.
(II) AWT applets offer a richer and easier-to-use user interface than Swing.
(III) AWT are stand-alone programs.

a). I only	b). II only	c). I and II only
d). I and III only	e). All I, II, and III	

7). Which of the following statements is ***correct*** regarding ***Java classes and objects***?

a). Java's primitive types are implemented as objects.
b). Methods that have a return type of void must return a value.
c). A constructor does not initialize an object immediately upon creation; therefore, it needs to be called.
d). The implicit return type of a class constructor is the class type itself.
e). Declaring two local variables with the same name inside the same or enclosing scopes is allowed.

8). Which of the following statements is ***correct***?

(I) The FileReader class creates a Reader that one can use to read the contents of a file.
(II) The File class is defined by java.io.
(III) FileOutputStream creates an OutputStream that one can use to write bytes to a file.

a). I only	b). II only	c). I and II only
d). I and III only	e). All I, II, and III	

9). Which of the following statements are ***correct*** regarding Java ***layout managers***?

(I) FlowLayout is governed by the container's component orientation.

(II) The CardLayout stores other layouts and has them hidden, ready to be activated when needed.

(III) GridLayout defines the number of rows and columns.

a). I only	b). II only	c). I and II only
d). I and III only	e). All I, II, and III	

10). Which of the following statements ***correctly*** describes ***procedural program design***?

a). It breaks the problem into a set of separate objects that perform.

b). The design is based on the idea that an event can cause a program to change from one known state to another.

c). Based on the idea that the data in a program is more stable than the processes involved.

d). The structure of the data is considered at the start, before all the high-level processes of the program have been defined.

e). It is based on the idea that the most important feature of a program is its processes.

11). What would be the exception that would be thrown for this code?

```
1 class NumberApp {
2     public static void main(String args[]) {
3             int c[] = {1},
4             d[]=null;
5             int a = c.length;
6             int b = 42 / a;
7             System.out.println(b +d.length); }}
```

a). ArithmeticException	b). IllegalAccessException	c). ArrayIndexOutOfBoundsException
d). NullPointerException	e). RuntimeException	

12). Which of the following statements is ***correct***?

a). ***switch*** is an example of an iterative statement.

b). ***for*** loop is an example of a selection statement.

c). A ***continue*** terminates the current method and jumps to the place after the function call.

d). A ***break*** statement jumps out of a loop and bypasses the loop condition.

e). A ***return*** statement jumps out of the current iteration of a loop.

13). What should replace the blank in the following code to compute the sum of even numbers in the ***numbers*** array?

```
public class ListApp {
    public static void main(String[] args) {
        int[] numbers = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
        int sum = 0;
        for (int i = 0; i < numbers.length; i++) {
            if (numbers[i] % 2 != 0)
                ---------------
            sum = sum + numbers[i];}
        System.out.println(sum);}}
```

a). skip	b). continue	c). break
d). exit	e). return	

14). Java's abstract classes

a). include abstract methods that must be implemented in all subclasses.
b). include abstract methods that have a body (implementation) in the abstract class.
c). can only have abstract method declarations.
d). can be used to instantiate objects through their constructors.
e). can inherit from multiple classes.

15). Which of the following statements is ***correct***?

a). Java exception is a built-in function that could be included in a piece of code.
b). Exceptions are runtime events that disrupt the normal flow of a program.
c). Program statements that need to be monitored for exceptions are in a "catch" block.
d). Any code that must be executed after a catch block completes is put in a "throw" block.
e). In Java to manually throw an exception, the "try" keyword is used.

For questions 16 to 20, fill in the blanks by selecting the correct answer.

16). _____ binds together code and data and ensures security from outside interference and misuse.

a). Encapsulation	b). Inheritance	c). Polymorphism
d). Abstraction	e). Cohesion	

17). _____ is used when multiple threads need access to a shared resource, ensuring that only one thread can access the resource at a time.

a). Extending Thread	b). Centralization	c). Synchronization
d). Deadlocks	e). Racing Thread	

18). _____ is mostly used to receive user input and parse them into primitive data types such as int, double and strings.

a). Scanner class	b). File class	c). Random class
d). Math class	e). StringBuilder class	

19). _____ method is called by println() when outputting a Throwable object.

a). getMessage()	b). toString()	c). printStackTrace()
d). getStackTrace()	e). getCause()	

20). _____ supports communication between the JDBC manager and the database driver.

a). JDBC Bridge	b). JDBC Connection Pool	c). JDBC URL
d). JDBC API	e). JDBC driver	

21). Which of the following statements is ***incorrect*** regarding ***packages***?

a). A package could store a class named ”list” without conflicting with another class named ”list” stored elsewhere.

b). Packages need to be imported into new class definitions.

c). The package is a visibility control mechanism.

d). Package defines class names that are exposed only to members of the same package.

e). The case of the class name must match the package name exactly.

22). Which one of the following code segments is ***correct*** with respect to Java generics?

a). class GenericArray <T> T[] arr = new T[10];

b). class Gen<T> T ob; Gen()

c). class Test<T> static T ob;

d). class GenericArray<T> Gen<T>[] array = new Gen<T>[10];

e). class Box<T> Box<String> stringBox = new Box<>();

23). What are the suitable replacements for (A), (B), and (C) respectively in the following code segment?

```
interface Sum {
    void meth1();}
class MyClass __(A)__ Sum {
    __(B)__ __(C)__ meth1() {}}
class ClassApp {
   public static void main(String arg[]) {
    MyClass ob = new MyClass();
    ob.meth1();}}
```

a). implements , public , static	b). extends , public , void	c). implements , public , void
d). extends , public , static	e). implements , private , void	

24). Which of the following can be used to develop desktop application interfaces?

(I) JDBC

(II) Servlet

(III) JavaFx

a). I only	b). II only	c). III only
d). II and III only	e). All I, II, and III	

For questions from 25 to 40, select the execution outputs of the given code segments.

25).

```
class Animal {
    void AB() {
        System.out.print("B");}
        }
abstract class Dog extends Animal {
    abstract void AB();}

class AnimalApp {
    public static void main(String args[]) {
    Animal b = new Animal();
    b.AB();}}
```

a). A	b). B	c). AB
d). BA	e). " "	

26).
```
public class NumApp {
    public static void main(String[] args) {
        int val1,val2,val6;
        String val3,val4,val5;
        val1 = 5;val2 = 2;
        val2 *= 3;

        val3 = Integer.toString(val1);
        val4 = Integer.toString(val2);
        val5=val3+val4;
        System.out.println(val5);}}
```

a). 5	b). 7	c). 11
d). 56	e). 58	

27).
```
public class CharApp {
    public static void main(String args[]) {
        char ch1 = 'X';
        ch1++;
        System.out.println(ch1);}}
```

a). XX	b). Y	c). X++
d). X2	e). ch1	

28).
```
public class IndexApp {
    public static void main(String[] args) {
        outer:
        for (int i = 0; i < 2; i++) {
            for (int j = 0; j < 2; j++) {
                System.out.print("i=" + i + "j =" + j +", ");
                if (i == 0 && j == 1)
                {
                    break outer;
                    }}}}}
```

a). i = 0, j = 0

b). i = 1, j = 1

c). i = 0 j = 0, i = 0 j = 1,

d). i = 0 j = 0, i = 0 j = 1, i = 1 j = 0,

e). i = 0 j = 0, i = 0 j = 1, i = 1 j = 0, i = 1 j = 1,

29).
```
public class HelloApp {
    public static void main(String[] args) {
        char ch = 'A';
        int count = 0;
        while (count < 5) {
            System.out.print(ch);
            ch += 2;
            count++;}}}
```

a). ABCDE	b). EDCBA	c). ACEGI
d). A	e). C	

30).
```
public class NewApp {
    public static void main(String[] args) {
        int p = 3;
        int q = 5;
        int result = p * q;
        result = (result % 2 == 0) ? result / 2 : result * 2;
        System.out.println(result);}}
```

a). 7	b). 30	c). 243
d). 121	e). 486	

31).
```
class A {
    A() {
    System.out.print("A");}}
class B extends A {
    B() {
    System.out.print("B");}}
class C extends B {
    C() {
    System.out.print("C");}}
class D extends B {
    D() {
    System.out.print("D");}}
class HelloApp {
    public static void main(String args[]) {
    D c = new D();}}
```

a). ABC	b). CBA	c). ABD
d). DBA	e). C	

32).
```
public class MathApp {
    public static void main(String args[]) {
        int x = 123456___789;
        System.out.println(x);}}
```

a). 123456789	b). 123456___789	c). 123_456_789
d). 123,456,789	e). 122667	

33).
```
public class MyApp {
    public static void main(String[] args) {
        int a = 6;
        int b = 2;
        a |= b;
        System.out.println(a);}}
```

a). 0	b). 2	c). 3
d). 4	e). 6	

34).
```
public class FoodApp {
    public static void main(String[] args) {
        String[] words = {"apple","banana","bananas","Banana"};
        for (int i = 0; i < words.length; i++) {
            for (int j = i + 1; j < words.length; j++) {
                if (words[i].compareTo(words[j]) > 0) {
                    String temp = words[i];
                    words[i] = words[j];
                    words[j] = temp;}}}
        for (String word : words) {
            System.out.print(word + " ");}}}
```

a). apple Banana banana bananas

b). apple banana bananas Banana

c). apple bananas banana Banana

d). Banana apple banana bananas

e). apple banana bananas Banana

35).
```
public class WordApp {
    public static void main(String[] args) {
        String str = "Hello, welcome to Java!";
        String subStr = str.substring(7);
        System.out.println(subStr);}}
```

a). w	b). Hello,	c). welcome
d). welcome to Java!	e). o Java!	

36).
```
class Calculator {
    int add(int a, int b) {
    return a/b;}}

class NewCal extends Calculator {
    int add(int a, int b) {
        int sum = super.add(a, b);
        return sum + 10; }}

public class HelloApp {
    public static void main(String[] args) {
        NewCal myCalculator = new NewCal();
        int result = myCalculator.add(10, 4);
        System.out.println(result); }}
```

a). 11	b). 12	c). 12.5
d). 14	e). 24	

37).
```
class App {
    int a=10;
    private int b=20;
    public int c =b*2;}

class OurApp {
    public static void main(String args[]) {
    App ob = new App();
    System.out.println( ob.a + " "+ob.c );}}
```

a). 10 20	b). 20 20	c). 10 20*2
d). 10 40	e). 10 b*2	

38).
```
enum Car {
    Tesla(250), Ford(200), BMW(240), Audi(230), Toyota(180);
    public int topSpeed;
    Car(int speed) {
        topSpeed = speed;}

    static int Cars250() {
        int count = 0;
        for (Car car : values()) {
            if (car.topSpeed - 100 > 120) count++;}
        return count;}}

class HelloApp {
    public static void main(String[] args) {
        System.out.println(Car.Cars250());}}
```

a). 1	b). 2	c). 3
d). 4	e). 5	

39).
```
enum Color {Red, Green}
class HelloApp {
    public static void main(String[] args) {
        Color[] c = Color.values();
        for (int i = c.length - 1; i >= 0; i--)
            System.out.print(c[i] + " " + c[i].ordinal()+" ");}}
```

a). Red 1 Green 2	b). Red 0 Green 1	c). Green 2 Red 1
d). Green 1 Red 0	e). Green 0 Red 1	

40).
```
import java.util.*;
class CodeApp {
    public static void main(String args[]) {
        List<Integer> list = new ArrayList<>();
        for (int i = 1; i <= 4; i++) {
            list.add(i);}
        int X = 1;
        for (int i : list) {X *= i;}
        System.out.println(X);}}
```

a). 10	b). 24	c). 1 2 3 4
d). 11	e). 25	

_____ ********** _____