

প্রশ্ন ৪৮. Write Output from below code:

```
#include <stdio.h>
int main(){
    char s[] = "Bangladesh Industrial and
    Technical Assistant Center";
    char *s1;
    s1=s;
    for(int i=0; i<10; i++){
        printf(" %c ", s[i]);
        ++s1;
    }
    printf("\n");

    for(int i=0; i<10; i++){
        printf(" %c ", s1[i]);
        ++s1;
    }
    return 0;
}
```

Output:

```
B a n g l a d e s h
n u t i l a d T c
```

Object Oriented Programming (C++/Java)

[**Syllabus:** BPSC CS: Introductory programming with C++/JAVA. Introduction to object oriented programming: Encapsulation, inheritance and polymorphism, Mechanic Language Programming, Template functions and classes multi-threads exceptions, Class and object.

NTRCA ICT: Topics include object-oriented programming concepts, such as classes, objects, methods, interfaces, packages, inheritance, encapsulation, and polymorphism.]

**অবজেক্ট অরিয়েন্টেড প্রোগ্ৰামিং
(Object Oriented Programming)**

প্ৰশ্ন ১. অবজেক্ট অরিয়েন্টেড প্রোগ্ৰামিং ল্যাংভেজ বলতে কী
উত্তৰঃ [NTRCA-2014, BOF-2018, BOF-SAE-2021]
উত্তৰঃ অবজেক্ট অরিয়েন্টেড বৈশিষ্ট্য সম্পৰ্ক যে কোন ভাষাকে অবজেক্ট
অরিয়েন্টেড প্রোগ্ৰামিং ল্যাংভেজ বলে। যেহেতু ইনহেরিটেস,
পলিমোফিজম, এনক্যাপ্সুলেশন ইত্যাদি।

প্ৰশ্ন ২. অবজেক্ট অরিয়েন্টেড প্রোগ্ৰামিং এর কিছু বৈশিষ্ট্য লিখ। [সহকাৰী
মেইল ইডঃ 10, BPDB-2004, BOF-2018, NTRCA-
2014, 16, 19, 2020, Uttara Bank-2019, BCC AP(2019), বাণিজ্য
মন্ত্ৰণালয়-২০২১-সহকাৰী প্ৰেমাবৰ্ণ (নন-কাউন্সেল), BOF-SAE-2021]

উত্তৰঃ অবজেক্ট অরিয়েন্টেড প্রোগ্ৰামিং এর বৈশিষ্ট্য গুলো হলো :

- Inheritance
- Polymorphism
- Encapsulation
- Data Hiding
- Static Binding
- Dynamic Binding
- Message Passing

এই ধাৰণাতলিগতি ছাড়াও আৱশ্যিক কিছু Terms রেখে যা অবজেক্ট-
অরিয়েন্টেড ডিজাইনে ব্যবহৃত হয়:

1. Coupling
2. Cohesion
3. Association
4. Aggregation

প্ৰশ্ন ৩. কয়েকটি অবজেক্ট অরিয়েন্টেড প্রোগ্ৰামিং ল্যাংভেজের নাম লিখ।
উত্তৰঃ কয়েকটি অবজেক্ট অরিয়েন্টেড প্রোগ্ৰামিং ল্যাংভেজের নাম
নিম্নোক্তগুলি

- a. Python
- b. Turbo Pascal
- c. C++
- d. Java
- e. Hyper Talk
- f. Visual Basic
- g. Visual C

প্ৰশ্ন ৪. ট্ৰান্সিশনাল প্রোগ্ৰামিং এর অসুবিধাসমূহ লিখ।

উত্তৰঃ

- বড় বড় প্ৰোগ্ৰামেৰ ক্ষেত্ৰে কোন ডাটা ব্যবহৃত হচ্ছে তা Identify কৰিব।
- সিস্টেমেৰ Security দ্বাৰা পায়।
- এ ধাৰনেৰ প্ৰোগ্ৰামিং-এ কুল কুটি নিৰ্ধাৰণ কৰা কৰিব।
- এই পদ্ধতিতে বড় বড় প্ৰোগ্ৰাম লিখতে পৰিৱে �data abstraction,
code reusability ইত্যাদি বিষয়ে সমস্যা পড়তে হৈ।
- ট্ৰান্সিশনাল প্রোগ্ৰামিং এৰ কোডিং কৰা বেশ কষ্টকৰ।

প্ৰশ্ন ৫. প্ৰসিডিউল অরিয়েন্টেড প্রোগ্ৰামিং ও অবজেক্ট অরিয়েন্টেড
প্রোগ্ৰামিং এৰ পাৰ্থক্য লিখ। [ICT D-AP-2021]

উত্তৰঃ

প্ৰসিডিউল অরিয়েন্টেড প্রোগ্ৰামিং | অবজেক্ট অরিয়েন্টেড প্রোগ্ৰামিং

১। High Level Language ব্যবহাৰ না কৰে	১। POP-এ বিদ্যমান অসুবিধা সমূহ সমাধানেৰ নিমিত্তে যে প্ৰয়াম রচিত তাকে প্ৰসিডিউল অরিয়েন্টেড প্রোগ্ৰামিং বলে।
২। মূল প্ৰোগ্ৰামকে কৃততলো module এ ভাগ কৰা হায়।	২। মূল প্ৰোগ্ৰামকে কৃততলো অবজেক্টে ভাগ কৰা হায়।
৩। এতে ডাটা অপেক্ষা ফাংশনেৰ ভৰ্তু বৈশী।	৩। এতে ফাংশন অপেক্ষা ডাটাৰ ভৰ্তু বৈশী।
৪। এই ধাৰনেৰ প্ৰোগ্ৰামিং টপ ডাউন পদ্ধতিতে কৰা হায়।	৪। এই ধাৰনেৰ প্ৰোগ্ৰামিং বটম আপ পদ্ধতিতে কৰা হায়।
৫। সাধাৰণ নিয়মে এক ফাংশন অন্য ফাংশনেৰ সাথে সংযোগ রাখা কৰে।	৫। ম্যাসেজ পাসিং এৰ মাধ্যমে এক অবজেক্টে অন্য অবজেক্টেৰ সাথে সংযোগ রাখা কৰে।
৬। এই ধাৰনেৰ প্ৰোগ্ৰামিং-এ কুল-কুলি নিৰ্ধাৰণ কৰা কৰিব।	৬। এই ধাৰনেৰ প্ৰোগ্ৰামিং-এ কুল-কুলি নিৰ্ধাৰণ কৰা সহজ।
৭। এই পদ্ধতিতে প্ৰোগ্ৰামেৰ মেইনটেনেন্স কৰা অনেক সহজ।	৭। এই পদ্ধতি প্ৰোগ্ৰামেৰ মেইনটেনেন্স অনেক সহজ।
৮। উদাহৰণঃ Fortarn, Pascal, Basic, ইত্যাদি।	৮। উদাহৰণঃ C++, Java

প্ৰশ্ন ৬. অবজেক্ট কাকে বলে?

উত্তৰঃ অবজেক্ট অরিয়েন্টেড প্রোগ্ৰামিং এৰ ক্ষেত্ৰে কোন type এৰ
Variable সমূহকে এক একটি অবজেক্ট বলে।

Syntax:

Class_Name variable_Name;

Ex: Student student;

প্ৰশ্ন ৭. সি++ কে অবজেক্ট অরিয়েন্টেড ল্যাংভেজ বলা হায় কেন?

উত্তৰঃ যেহেতু সি++ ভাষাৰ অবজেক্ট অরিয়েন্টেড ল্যাংভেজেৰ সকল
জৰুৰী বৈশিষ্ট্যসমূহ (যেহেতু কুস, ইনহেরিটেস, পলিমোফিজম)
বিদ্যমান তাই C++ কে অবজেক্ট অরিয়েন্টেড ল্যাংভেজ বলে।

প্ৰশ্ন ৮. সি++ কে হাইত্ৰিড প্রোগ্ৰামিং ল্যাংভেজ বলা হায় কেন?

উত্তৰঃ সি++, সি ভাষাৰ মত একদিকে যেহেতু প্ৰসিডিউল প্ৰোগ্ৰামিং এৰ
সুবিধা প্ৰদান কৰে এবং একই সাথে অবজেক্ট অরিয়েন্টেড প্ৰোগ্ৰামিং
পদ্ধতিৰ ব্যবহাৰও অনুমোদন কৰে, তাই C++ কে হাইত্ৰিড প্ৰোগ্ৰামিং
ল্যাংভেজ বলে।

প্ৰশ্ন ৯. সি++ প্ৰোগ্ৰামেৰ বেসিক স্ট্রাকচাৰ বৰ্ণনা কৰ। [DPDC-2018]

উত্তৰঃ সি++ প্ৰোগ্ৰামেৰ Structure নিম্নোক্তঃ

Include File
Class Declaration
Member Function Definition
Main Function()

প্ৰশ্ন ১০. কয়েকটি সি++ কম্পাইলাৰেৰ নাম লিখ।

উত্তৰঃ কয়েকটি বহুল ব্যবহৃত সি++ কম্পাইলাৰেৰ নাম হল ANSI
C++, Turbo C++, Borland C++, visual C++, gcc(g++) C++ ইত্যাদি।

প্ৰশ্ন ১১. সি++ ব্যবহৃত কয়েকটি স্পেশাল অপাৱেটৱেৰ নাম লিখ।
উত্তৰঃ সি++ এ ব্যবহৃত কয়েকটি স্পেশাল অপাৱেটৱ হলঃ ইনসাৰ্চ
অপাৱেটৱ (<<), এক্স্ট্ৰাকশন অপাৱেটৱ (>>), কোপ ৰেজুলুশন
অপাৱেটৱ (: :), new, delete, endl, ends, ইত্যাদি।

প্ৰশ্ন ১২. কুস কী? কুস ঘোষনাৰ ফৰম্যাট বা Syntax লিখ।
উত্তৰঃ কুস হচ্ছে এক ধাৰনেৰ ইউজাৰ ডিফাইনড ডাটা টাইপ। যাতে একই
শ্ৰেণিৰ এক বা একাধিক অবজেক্ট থাকে। এটি প্ৰয়ামে Build in data
type এৰ ন্যায় কাজ কৰে। সাধাৰণ অৰ্থে, একই ধাৰনেৰ কৃততলোৱে
অবজেক্ট সমৰ্থিত নামকে কুস বলে।

Syntax:

```
class Class_Name{
    Access specifier;
    Data members;
    Member Functions(){}
};
```

প্ৰশ্ন ১৩. কুস এবং অবজেক্ট এৰ মধ্যে পাৰ্থক্য লিখ।

উত্তৰঃ কুস এবং অবজেক্ট এৰ মধ্যে পাৰ্থক্য নিম্নে দেওয়া হলো:

কুস	অবজেক্ট
কুস হচ্ছে এক ধাৰনেৰ ইউজাৰ ডিফাইনড ডাটা টাইপ। যাতে একই শ্ৰেণিৰ বা গোৱেৰ এক বা একাধিক অবজেক্ট থাকে।	Object হচ্ছে instance of a class।
Class হচ্ছে logical entity।	Object হচ্ছে physical entity
মেমৰি space ব্ৰাঞ্চ কৰা হয়, না, যখন এটি তৈৰি কৰা হয়।	মেমৰি space ব্ৰাঞ্চ কৰা হয়, যখন এটি তৈৰি কৰা হয়।
কুস একবাৰ তৈৰি কৰা হয়।	এটি প্ৰয়োজন মত অনেক বাৰ তৈৰি কৰা হয়।

Real World Example of Objects & Classes: কাৰ

নামে একটা কুস আছে, তাৰ দুইটা কপি বানালাম Car1 এবং
Car2 নামে। Car Class টাইপেৰ দুইটা কপি হল দুইটা অজেক্ট।

Class: FordCar

Attributes

Color, Type, Model, Cylinder

Behavior

Start, Accelerate, Reverse, Stop

Object: FordCar1

Attributes

Color: Yellow

Type: Coupe

Model: Mustang

Cylinder: 6

Behavior

Start, Accelerate, Reverse, Stop

Object: FordCar2

Attributes

Color: Orange

Type: Coupe

Model: Focus

Cylinder: 4

Behavior

Start, Accelerate, Reverse, Stop

প্রশ্ন ১৮. মেধার ফাংশন ও মেধার তেরিহেল কাকে বলে?
উত্তর: ক্লাসের অভ্যন্তরে ডিক্রিয়ারকৃত ফাংশনসমূহকে মেধার ফাংশন
বলে। আর ক্লাসের অভ্যন্তরে ডিক্রিয়ারকৃত তেরিহেলসমূহকে মেধার
তেরিহেল বলে।

প্রশ্ন ১৯. মেধার ফাংশনকে কয়তবে define করা যায়?

উত্তর: মেধার ফাংশনকে দুইভাবে define করা যায়।
- outside the class definition
- inside the class definition

প্রশ্ন ২০. Base class & Derived class কাকে বলে?

উত্তর: কোন একটি ক্লাস যদি পূর্বে ডিফাইনকৃত অন্যকেন ক্লাসের
বৈশিষ্ট্য ইনহেরিট করে তবে যে ক্লাস থেকে ইনহেরিট করা হয় তাকে
Base class আর যে Class বৈশিষ্ট্য ইনহেরিট করে তাকে Derived
class বলে।

প্রশ্ন ২১. What is virtual machine? [BTEB-2005]

উত্তর: জাভা ভার্টুয়াল মেশিন সান মাইক্রোসিস্টেম-এর একটি সফটওয়্যার
যা বাইটকোড রান করতে সক্ষম। এটি জাভা বাইটকোডের একটি
প্রোগ্রামকে মেশিন ভাষায় পরিবর্তিত ও তা রান করে। এটি জাভা
সফটওয়্যার প্রার্টফর্মের কোটি একজিকিউশনের সহায়ক একটি উপাদান।

প্রশ্ন ২২. Pure Virtual Functions?

Pure virtual Functions হল এমন একটি virtual functions
যার কোন definition থাকে না।
তবে যে virtual keyword দিয়ে আর শেষ হয় =0 দিয়ে।

Syntax: virtual void Test() = 0;

প্রশ্ন ২৩. Virtual Base Class কাকে বলে?

উত্তর: Multiple inheritance এর অসম্ভবসাধা মূল্যবাননের জন্য যে
Class ব্যবহৃত হয় তাকে Virtual base class বলে।

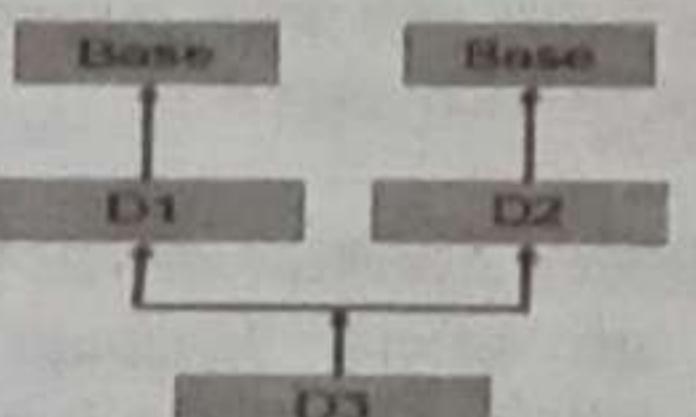


Fig: Virtual Base Class

প্রশ্ন ২৪. Structure & Class এর মধ্যে পার্থক্য লিখ।
উত্তর: Structure & Class এর মধ্যে পার্থক্য নিম্নরূপ:

Structure	Class
১: Structure হল এক ধরনের ইউজার ডিফাইনড ভাটা টাইপ।	১: ক্লাস হল স্ট্রাকচারের বর্ধিত রূপ।
২: এটি প্রোগ্রামে Built in data type এর নাম কাজ করে।	২: এটি প্রোগ্রামে Built in data type এর নাম কাজ করে না।
৩: এতে সিস্টেমে ব্যবহৃত ভাটার গোপনীয়তা ও বিশৃঙ্খলা	৩: এতে সিস্টেমে ব্যবহৃত ভাটার গোপনীয়তা ও বিশৃঙ্খলা

প্রশ্ন ২৫. ক্লাস প্রোগ্রাম কৈ করে?

ক্লাস প্রোগ্রাম:	কৈ করে:
১: Structure এ যে কোন সংখ্যক Variable ব্যবহার করা যায়।	১: ক্লাস যে কোন সংখ্যক Variable ব্যবহার করা যায়।
২: যে কোন প্রোগ্রামে Structure একটি থাকতে পারে।	২: প্রোগ্রামে এক বা একাধিক ক্লাস থাকতে পারে।

প্রশ্ন ২৬. ডাটা স্ট্রিম কৈ কৈ বুঝ?

উত্তর: কোন একটি ক্লাস যদি পূর্বে ডিফাইনকৃত অন্যকেন ক্লাসের
বৈশিষ্ট্য ইনহেরিট করে তবে যে ক্লাস থেকে ইনহেরিট করা হয় তাকে
প্রবাহকে ডাটা স্ট্রিম (Data Stream) বলে।

প্রশ্ন ২৭. কনসোল I/O কৈ? ইয়ে কৈ প্রকার ও কৈ কৈ?

উত্তর: C++ প্রোগ্রামে ইনপুট আউটপুট অপারেশন সম্পর্ক করার জন্য যে
সকল Statement ব্যবহার করা হয় তাদেরকে কনসোল I/O বলে।
ইয়ে দুই প্রকার। যথা-

- a. Formatted I/O
- b. Unformatted I/O

প্রশ্ন ২৮. Unformatted input operation এ ব্যবহৃত ফাংশন
সমূহের নাম লিখ।

উত্তর: Unformatted input operation এর জন্য সাধারণত cin,
get(), read(), getline(), ignore(), peek(), putback()
() ইত্যাদি ফাংশন ব্যবহৃত হয়।

প্রশ্ন ২৯. ডট অপারেটর(.) কাকে বলে?

উত্তর: সাধারণত iostream stream এর সাথে Formatted &
Unformatted input-output operation এর সংযোগ ঘটাতে
যে ধরনের অপারেটর ব্যবহার করা হয় তাকে ডট অপারেটর(.) বলে।

প্রশ্ন ৩০. সি++ এ ব্যবহৃত ৫টি কী-ওয়ার্ডের নাম লিখ যেতো সি
ভাষায় অর্থীভূত নয়।

উত্তর: ডুমাত সি++ এ ব্যবহৃত ৫টি কী-ওয়ার্ড হলঃ catch, class,
new, delete, private,public,protected ইত্যাদি।

প্রশ্ন ৩১. What is Binding? বিভিন্ন প্রকার বাইভিং সম্পর্কে লিখ।

Binding: binding হল identifiers এর নাম কে মেশিন কোড এ
কপার্সের একটি প্রসেস।

Early binding/ Static: compile time function এ কোন
function call হয় তা জানা যায়।

Late Binding/dynamic binding/ Runtime Binding:
run time এ কোন function call হয় তা জানা যায়।

কিপ্রয়ার্টি দিয়ে : এবং যে সব ক্লাস এই ইন্টারফেস মেনে চলে তারা
এই ইন্টারফেসকে implement করে।

ইন্টারফেসের ডিফাইন করা হয় সাধারণভাবে ক্লাসের মত করেই;
ইন্টারফেসের মেটডগুলোর অধুনাতমোর (কি কি প্যারামিটার নেয়া)
ডিফাইন করে দেওয়া হয় কিন্তু এই মেটডগুলোর বডি ডিফাইন করা হয় না।

মেন:

```
interface A {
    public void show();
}
```

Example:

```
Public abstract interface IManipulation{
//Interface declaration
Public abstract void add(); //method declaration
public abstract void subtract();
}
```

- interface এর ভিতরে সকল মেথড ইন্টারফেসে public abstract void।
- interface এর ভিতরে সকল variables internally public static final (constants).
- ক্লাস interface কে implement করতে পারে কিন্তু extends করতে পারে না।
- Class যে ইন্টারফেস কে ইম্প্রিমেট করছে সেই ইন্টারফেসে ঘোষিত
সকল মেথডের ইম্প্রিমেটেশন দিতে হবে।

```
public class Manipulation implements
IManipulation{ //Manipulation class uses the
interface
Public void add(){
.....
}
Public void subtract(){
.....
}}
```

প্রশ্ন ৩৬. Interface কি final করা যায়?

উত্তর: না, কারণ একটি ইন্টারফেস অন্য ক্লাসের ধারা ইম্প্রিমেট করতে
হয়। ফাইনাল হলে এটিকে ইম্প্রিমেট করা যাব না।

প্রশ্ন ৩৭. Abstract class and interface এর মধ্যে পার্থক্য
কৈলো লিখ?

উত্তর:

Abstract class	Interface
abstract class এ non- abstract methods(method without body) থাকতে পারে,	অধুনাত abstract methods থাকে

instance variables থাকতে পারে	instance variables থাকতে পারে না
constructor থাকতে পারে	Constructor থাকতে পারে না
static methods থাকতে পারে	static methods থাকতে পারে না
একটি abstract class extend করা যাব।	একাধিক interfaces কে implement করা যাব।
abstract class interface এর implementation দিতে পারে।	Interface abstract class implementation দিতে পারে না।
abstrac class declare করার জন্য abstract keyword ব্যবহার করা হয়।	interface declare করার জন্য interface keyword ব্যবহার করা হয়।
একটি abstract class অন্য জাভা class কে extends করতে পারে এবং একাধিক জাভা ইন্টারফেস implement করতে পারে।	একটি ইন্টারফেস কেবল অন্য জাভা ইন্টারফেসকে extends করতে পারে।
extends কীওয়ার্ড ব্যবহার করে একটি abstract class কে একটি করা মেটে পারে	একটি ইন্টারফেস ক্লাস interface কীওয়ার্ড প্রয়োগ ব্যবহার করে implement করা মেটে পারে।
একটি জাভা abstract class এ private, protected, ইত্যাদি ক্লাস থাকতে পারে	public by default.
Example: public abstract class Shape{ public abstract void draw(); }	Example: public interface Drawable{ void draw(); } Ex: class WithFriend{ int i; public: friend void fun(); // Global function as friend }; void fun(){ WithFriend wf; wf.i=10; cout << wf.i; } int main(){ fun(); }

প্রশ্ন 38. What is Static functions?

উত্তর যখন একটি ক্লাস এর মেদার কে static হিসাবে declare করা হয় তখন সকল object এর জন্য , member function এর উদ্বোধ একটি কপি তৈরি হয়।

Ex:

```
class X{  
public:  
    static void f(){};  
};  
int main(){  
    X::f();  
}
```

প্রশ্ন 39. What is Inline function?

উত্তর যখন কোন মেদার function এর definition ক্লাস এর ভিতর লিখা হয় তাকে Inline functions বলে।

```
#include <iostream>  
using namespace std;  
inline int Max(int x, int y){  
    return (x > y)? x : y;  
}  
  
int main(){  
cout << "Max (20,10): " << Max(20,10) << endl;  
cout << "Max (0,200): " << Max(0,200) << endl;  
cout << "Max (100,1010): " << Max(100,1010) << endl;  
return 0;  
}
```

প্রশ্ন 40. Inline member function এর সুবিধা এবং অসুবিধা সমূহ লিখ।

উত্তর ক্লাসের ইনলাইন মেদার ফাংশন এর ক্ষেত্রে যখন কোন অবজেক্ট ঘোষণ করা হয় তখন প্রতিটি অবজেক্ট এর জন্যেই মেদার ফাংশন এর একটি কপি তৈরী হয়। ফলে প্রোগ্রাম রান হবার সময় জম্প ইন্স্ট্রুকশন এর জন্য যে সব লাগে তা বেচে যায়। ফলে প্রোগ্রাম দ্রুত রান করে। কিন্তু অসুবিধা হচ্ছে যে প্রতিটি অবজেক্ট এর জন্য ইনলাইন মেদার ফাংশন এর একটি কপি তৈরী হবার ফলে মেমরি এর অপচয় হয়। অর্থাৎ একেজে মেমোরী বেশী লাগে।

প্রশ্ন 41. What is Friend function/Class? [ECS-2012]

উত্তর Friend function/Class কেন Class এর member নয়। Class এর private variable কেনো সাধারণত বাইরে থেকে ব্যবহার করা যাব না কিন্তু Friend function সেটা করতে পারে।

Example :

```
class WithFriend{  
int i;  
public: friend void fun();  
// Global function as friend  
};  
void fun(){  
WithFriend wf;  
wf.i=10;  
cout << wf.i;  
}
```

প্রশ্ন 42. মেথড ব্যবহারের সুবিধা সমূহ লিখ।

উত্তর মেথড ব্যবহারের সুবিধাসমূহ নিচেরপৰ্য

- মেথড ক্লাসের ভাটা সরাসরি কেট এক্সেস করতে পারে না।
- মেথড ব্যবহার করার ফলে প্রোগ্রামের কোন অংশে মূল প্রোগ্রামে কার লিখার প্রয়োজন হয় না।
- প্রোগ্রামের সাইজ ছোট হয়।
- প্রোগ্রাম লিখতে অপেক্ষাকৃত কম সময় লাগে।
- প্রোগ্রাম পড়ে বুকা ও সংশোধন করা সহজ হয়।

প্রশ্ন 43. Method/Function ওভারলোডিং বলতে কী বোঝায়?
[Bangladesh Bank-2014, IP-2015, Bangladesh Bank-2017, BPSC-2019, 40th bcs]

উত্তর দুই বা ততোধিক Method এর নাম একই হলে তাকে Method ওভারলোডিং বলে।

Method overloading এর পদ্ধতি :

- Arguments এর সংখ্যা পরিবর্তন করে।
- Argument এর types পরিবর্তন করে।

একটি ক্লাসে একই নামে একাধিক মেথড তৈরি করা যাব। তবে একই নামের ঘোষণানো দুটি মেথডের সবগুলো প্যারামিটার ও রিটুর্ন টাইপ একই হতে পারবেনো। মেথড কল করার সময় প্যারামিটার আর রিটুর্ন টাইপ মেঝে বুঝে নেয়া হয় কোন মেথডটিকে কল করা হয়েছে। একে ওভারলোড মেথড বা মেথড ওভারলোডিং বা ফাংশন ওভারলোডিং বলা হয়।

- Same method name
- Different argument type
- May have different return types

Ex:

```
#include <iostream>  
using namespace std;  
int operate (int a, int b){  
    return (a*b);  
}  
double operate (double a, double b){  
    return (a/b);  
}  
int main (){  
    int x=5,y=2;  
    double n=5.0,m=2.0;  
    cout << operate (x,y) << '\n';  
    cout << operate (n,m) << '\n';  
    return 0;  
}
```

প্রশ্ন 44. Function Overriding বলতে কী বোঝায়?

[Bangladesh Bank-2014].

উত্তর যখন base class কে derived class এ inherit করা হয় এবং base class এর কোন function derived class এ ব্যবহার করা হয় তখন একে Function Overriding বলা হয়।

Example of Function Overriding:

```
class Base{  
public: void show(){  
    cout << "Base class";  
}  
};  
class Derived:public Base{  
public:  
    void show(){  
        cout << "Derived Class";  
    }  
};
```

প্রশ্ন 45. ফাংশন ওভারলোডিং এর সীমাবদ্ধতা লিখ।

উত্তর ফাংশন ওভারলোডিং এর সীমাবদ্ধতা নিচেরপৰ্য

- এতে ব্যবহৃত ফাংশনগুলোর ক্ষেত্রে return type এর কোন চুম্বিকা নেই।
- একই ফাংশন একাধিক ফাংশন একটি কমন ঘোষনার মাধ্যমে ব্যবহার করা যাব না।
- প্রতিটি ফাংশন আলাদা ডিক্লেয়ার ও ডিফাইন করতে হয়।
- সব ধরনের কম্পাইলার ফাংশন ওভারলোডিং প্রক্রিয়া অনুমোদন করে না।

প্রশ্ন 46. Method Overloading and Overriding এর মধ্যে পার্শ্বক্ষেত্রে লিখ। [NTRCA-2011, বিভিন্ন ম্যানস -২০২০]
সহকারী মেইটেনেন্স ইন্সিলার (সম-ক্যারার)

উত্তর:

- ওভারলোডিং হয় কম্পাইল টাইমে অন্য সিকে ওভাররাইতিং হয় রান টাইমে।
- স্ট্যাটিক মেথডকে ওভারলোড করা যাব অন্য সিকে স্ট্যাটিক মেথডকে ওভাররাইতিং করা যাব না।
- ওভারলোডিং হয় একি ক্লাসে অন্য সিকে ওভাররাইতিং হয় ডিজ ক্লাসে (বেস এবং চাইন্স ক্লাস)
- ওভারলোডিং এর ক্ষেত্রে স্ট্যাটিক বাইডিং ব্যবহার করা হয় কিন্তু ওভাররাইতিং এ ডাইনামিক বাইডিং ব্যবহার করা যাব।
- ওভাররাইতিং থেকে ওভারলোডিং এর পারফর্মেন্স ভালো।
- প্রাইভেট এবং ফাইনাল মেথডকে ওভারলোড করা যাব কিন্তু ওভাররাইতিং করা যাব না।
- মেথড ওভারলোডিং এর ক্ষেত্রে আর্গুমেন্ট সংখ্যা অথবা আর্গুমেন্ট টাইপ ডিজ হতে হয় কিন্তু ওভাররাইতিং এ সমান থাকে।
- তখন ওভারলোড এর মেথডের রিটুর্ন টাইপ পরিবর্তন করে মেথড ওভারলোডিং সম্পাদন করা যাব না কিন্তু ওভাররাইতিং এ রিটুর্নের ধরণ অবশ্যই একই রকম হবে

প্রশ্ন 47. Private methods কি ওভাররাইত করা যাব?

উত্তর না, আমরা প্রাইভেট মেথডগুলিকে ওভাররাইত করতে পারি না কারণ প্রাইভেট মেথডের ক্লাসের মধ্যে সীমাবদ্ধ এবং আমরা সেগুলি ক্লাসের বাইরে আর্জেস করতে পারি না।

প্রশ্ন 48. আমরা কি সাবক্লাসে ওভাররাইত মেথডের কোপ পরিবর্তন করতে পারি?

উত্তর হ্যাঁ, আমরা সাবক্লাসে ওভাররাইত মেথডের কোপ পরিবর্তন করতে পারি। তবে, আমাদের অবশ্যই লক্ষ্য করতে হবে যে আমরা মেথডের অ্যারেসেসেগুলো ত্রাস করতে পারি না। মেথডের অ্যারেসেসিবিলিটি পরিবর্তন করার সময় নিম্নলিখিত বিষয়গুলি অবশ্যই ধোলা রাখা উচিত।

- প্রাইভেট কে protected, public, or default এ পরিবর্তন করা যাব
- protected কে public or default এ পরিবর্তন করা যাব
- default কে public এ পরিবর্তন করা যাব
- পাবলিক সবসময় পাবলিক হবে

প্রশ্ন ৪০. Operator Overloading কাকে বলে? [NTRCA-2014, IP-2014, 40th bcs]

উভয় কোন অপারেটর এর Built in যে কাজ করার ক্ষমতা থাকে তার মধ্যে সাড়তি কিছু কাজ করার ক্ষমতা তৈরীর প্রক্রিয়াকে অপারেটর উভারলেডিং বলে।

প্রশ্ন ৪১. অপারেটর কাজ করার পদ্ধতি এবং সাধারণ নিয়মগুলো লিখ :

উভয় অপারেটর কাজ করার পদ্ধতি এবং সাধারণ নিয়মগুলো নিচেরস্বত্ত্বে

> কেবল বিশ্বাস অপারেটরের জন্যই কাজ করার পদ্ধতি।

স্থূল কোন অপারেটর তৈরী করা যায় না।

> বিশ্বাস অপারেটর সম্মত প্রস্তুতির কাজ করা যায় না।

> উভারলেডের অপারেটর উভয়ই Class object এর ক্ষেত্রেই কাজ করে।

> উভারলেডের অপারেটরের মধ্যে Custom data type এর সম্মত ১টি Operand থাকতে হবে।

> কাটকগুলো অপারেটর এবং উভারলেড করার জন্য Friend Function ব্যবহার করা যায় না।

> বাইনারি অপারেটর Value return করার ক্ষেত্রে argument পরিবর্তন করা যায় না।

> Unary operator সম্মত জন্য প্রাচীক কোন argument প্রয়োজন হয় না।

প্রশ্ন ৪২. কোন Operators কলো overloaded হয় না?

- > scope operator - ::
- > sizeof()
- > member selector - .
- > member pointer selector - *
- > ternary operator - ?:

প্রশ্ন ৪৩. What is Namespace?

Ans: Namespace হল identifier এর ধারক।
তিনি প্রতিক্রিয়া করার ক্ষমতা আছে।

- Scope Resolution
- The using directive
- The using declaration

প্রশ্ন ৪৪. উভয় স্ট্যাটিক কিওয়ার্ড ব্যবহার করে মেরুকে কি উভারলেড করা যায়?

উত্তর : না, আমরা শুধু স্ট্যাটিক স্ট্যাটিক প্রয়োগ করে মেরুক উভারলেড করতে পারি না (প্রারম্ভিক সংখ্যা এবং পরমের একই)। নিম্নলিখিত উদাহরণ বিবেচনা করুন :

```
public class Animal {
    void consume(int a) {
        System.out.println(a+" consumed!!");
    }
    static void consume(int a) {
        System.out.println("consumed static "+a);
    }
}
```

public static void main (String args[]) {

```
Animal a = new Animal();
a.consume(10);
Animal.consume(20);
}
```

Output:

Animal.java:7: error: method consume(int) is already defined in class Animal
 static void consume(int a)

Animal.java:15: error: non-static method
consume(int) cannot be referenced from a static
context

Animal.consume(20);

2 errors

প্রশ্ন ৪৫. Main() method কি উভারলেড করা যায়?

উত্তর : হ্যাঁ, মেরুত উভারলেড করার ক্ষেত্রে জাত প্রয়োজন আবার
কোন স্থানক মেইন মেরুত থাকতে পারে।

প্রশ্ন ৪৬. Method overriding কি ?

উত্তর : হ্যাঁ একটি সামুদ্রিক ইতিহাসে তার প্রাচীন তাস হরা
ইন্সুলেটিকুল কোন মেরুতের স্পেসিফিক ইন্সুলেটিশন করে তাসে
তাকে মেরুত উভারলাইচ করে। এটি রান্টাইম প্লিমারিফিজ এবং
ইন্টারফেস মেরুত ইন্সুলেটিশনের জন্য ব্যবহৃত হয়।

মেরুত উভারলাইচ এর নিয়ম:

- o মেরুতের নাম প্যারেট তাসের মতো একই নামে থাকতে হবে।
- o মেরুতের signature প্যারেট তাসের মতো একই
signature থাকতে হবে।
- o সুষ্ঠু তাসের মধ্যে is-A relationship থাকতে হবে।

প্রশ্ন ৪৭. Static method কি উভারলাইচ করা যায়?

উত্তর : হ্যাঁ, আমনি স্ট্যাটিক মেরুতের উভারলাইচ করতে পারবেন না কারণ
তার তাসের অর্থনীতি, অবস্থার না।

প্রশ্ন ৪৮. Overloaded method কি উভারলাইচ করা যায়?

উত্তর : হ্যাঁ

প্রশ্ন ৪৯. What is the main difference between an array and a class? When would you use a class instead of an array? [বর্ষা সম্মত ও বৈশিষ্ট্য কর্তৃপক্ষের অন্তর্ভুক্ত খবরে-২০১৮ ইন্স্ট্রুক্টর (কম্পিউটার)]

উত্তর : আরে একই টাইপের একাধিক তালু স্টোর করে। তাস
অবস্থার জন্য টেইপলেট ডিজাইন করে, যাহাতে বিভিন্ন টাইপের
জাতীয় বিভিন্ন নামে একাধিক বার স্টোর করা যায়। সাধারণত তাসে
কাশেন্স/ মেরুত থাকে, অ্যারেতে থাকেনা। যখন আমাদের বিভিন্ন
টাইপের জাতীয় নিরে কাজ করার সরকার হবে, তখন আমরা আরে
প্রিমের তাস ব্যবহার করি।

Inheritance

প্রশ্ন ১. Inheritance কাকে বলে? [Islami Bank-2019, NTRCA-2014, PSC-2017]

উভয় কোন অপারেটর এর Built in যে কাজ করার ক্ষমতা থাকে তার
মধ্যে সাড়তি কিছু কাজ করার ক্ষমতা তৈরীর প্রক্রিয়াকে Inheritance বলে।

class Person {

....

};

class MathsTeacher : public Person {

....

};

class Footballer : public Person {

....

};

void sleep() {

cout << "I can sleep!" << endl;

}

// derived class

class Dog : public Animal {

public:

void bark() {

cout << "I can bark! Woof woof!" << endl;

}

};

int main() {

// Create object of the Dog class

Dog dog1;

// Calling members of the base class

dog1.eat();

dog1.sleep();

// Calling member of the derived class

dog1.bark();

return 0;

}

প্রশ্ন ২. Inheritance এর সুবিধাগুলো কি কি ?

উত্তর: Inheritance এর সুবিধা গুলো:

- Code reusability
- Increased reliability
- Code sharing
- Consistency of interface
- Rapid prototyping.

প্রশ্ন ৩. Inheritance কত প্রকার ও কি কি?

উভয় ইনহেরিট্যাল ও একার : যথা

(ক) পাবলিক ইনহেরিট্যাল :

> Base তাসের পাবলিক মেরুতগুলো Derived তাসের পাবলিক
মেরুত হয়ে যাবে।

> Base তাসের প্রোটোটাইপ মেরুতগুলো Derived তাসের
প্রোটোটাইপ মেরুত হয়ে যাবে।

> Base তাসের প্রাইভেট মেরুতগুলো Derived তাসের প্রাইভেট
স্যাম্পলি একেন করতে পারবে না। তবে পাবলিক বা প্রোটোটাইপ
কোনো মেরুতের সাহায্যে পরোক্ষভাবে একেন করতে পারবে।

(খ) প্রোটোটাইপ ইনহেরিট্যাল :

> Base তাসের পাবলিক এবং প্রোটোটাইপ মেরুতগুলো Derived
তাসের প্রোটোটাইপ মেরুত হয়ে যাবে

(গ) প্রাইভেট ইনহেরিট্যাল :

> Base তাসের পাবলিক এবং প্রোটোটাইপ মেরুতগুলো Derived
তাসের প্রাইভেট মেরুত হয়ে যাবে (Access Specifier)

// C++ program to demonstrate inheritance

#include <iostream>

using namespace std;

// base class

class Animal {

public:

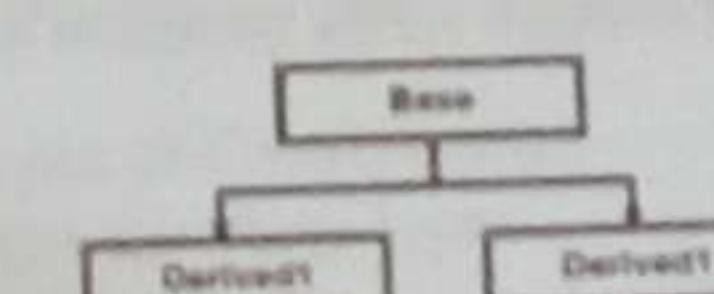
void eat() {

cout << "I can eat!" << endl;

}

প্রশ্ন ৫. Hierarchical Inheritance কাকে বলে?

উভয় যখন একটি আর Base class এতে দুই বা ততোধিক derived
class প্রস্তুত করা হয় তবে সেই প্রক্রিয়াকে হায়ারারিকাল ইনহেরিটেন্স
বলে।

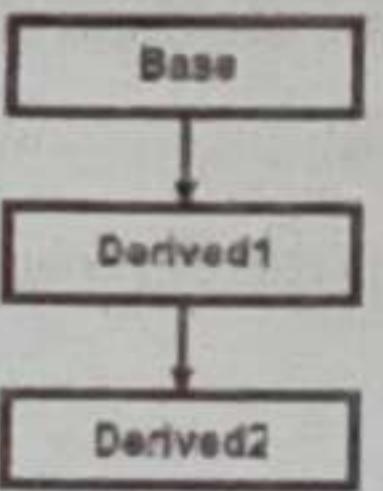


For example:

```
class A {
    .....
}
class B : A{
    .....
}
class C : A{
    .....
}
```

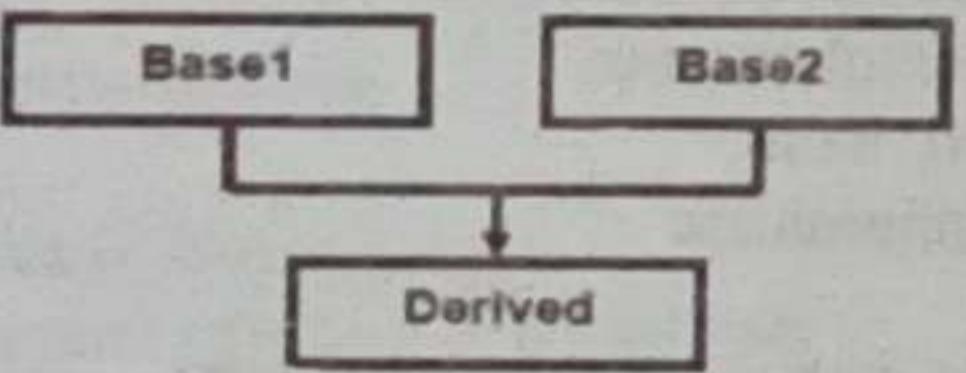
প্রশ্ন ৭. Multi Level Inheritance কাকে বলে?

উত্তরঃ যখন কোন Derived class অপর কোন derived class এর এক বা একাধিক বৈশিষ্ট্য inherit করে তখন এই প্রক্রিয়াকে Multilevel inheritance বলে।



প্রশ্ন ৮. Multiple Inheritance কাকে বলে?

উত্তরঃ যখন কোন Derived class দুই বা ততোধিক Base class থেকে দুই বা ততোধিক বৈশিষ্ট্য ইনহেরিট করে তবে সেই প্রক্রিয়াকে Multiple Inheritance বলে।



■ Why are the following two overloaded functions inherently ambiguous? [BCS-38th]

উত্তরঃ মাল্টিপ্ল ইনহেরিটেন্স এর ক্ষেত্রে ইনহেরিট ফাংশন ambiguous হয়। কারণ মাল্টিপ্ল ইনহেরিটেন্সের ক্ষেত্রে একটি বড় সমস্যা হলো ডায়মন্ড সমস্যা (Diamond Problem)। একটি উদাহরণ দেওয়া যাক, মনে করা যাক, একটি ক্লাস A, একে B ও C এক্সেন্ট করে। আবার আরেকটি ক্লাস D যা কিনা B ও C কে এক্সেন্ট করে। এখন যদি A এর কোনো একটি মেথডকে B ও C দুটিই উভাররাইট করে এবং D থেকে সেই মেথডটি কল করা হয়, তাহলে সেটি কোন ক্লাসের মেথডকে কল করবে? B না C? সমস্যাটি নিচ্ছি বোঝা যাচ্ছে। জাভা ডিজাইনাররা এরকম সমস্যার মধ্য দিয়ে যেতে চায়নি। তাই জাভাতে মাল্টিপ্ল ইনহেরিটেন্স নেই।

তাই জাভা ডিজাইনাররা জটিলতা ক্রান্ত করতে এবং ভাষা সরল করতে মাল্টিপ্ল ইনহেরিটেন্স জাভাতে সমর্থন করেনি।

```
class A{
    void msg(){System.out.println("Hello");}
}
```

```
class B{
    void msg(){System.out.println("Welcome");}
}
```

```
class C extends A,B{//suppose if it were
```

```
public static void main(String args[]){
    C obj=new C();
    obj.msg(); //Now which msg() method would be
    invoked?
}
```

Compile Time Error

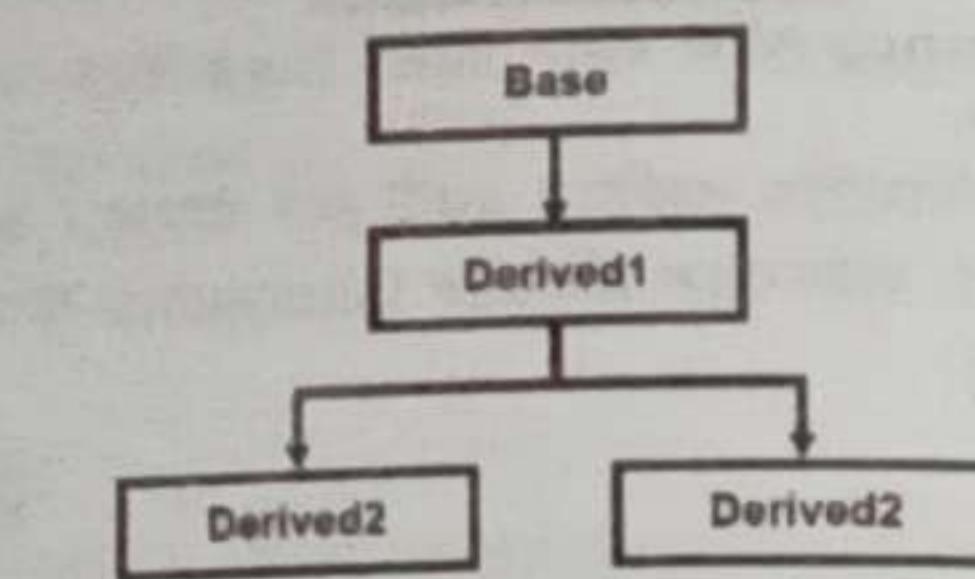
প্রশ্ন ৯. Hybrid Inheritance কাকে বলে?

উত্তরঃ যদি কোন derived class একই সাথে মাল্টিপ্ল ও মাল্টিলেভেল উভয় উপায়ে বিভিন্ন base class থেকে দুই বা ততোধিক বৈশিষ্ট্য ইনহেরিট করে তবে সেই প্রক্রিয়াকে হাইব্রিড ইনহেরিটেন্স বলে।

প্রশ্ন ১১. কত উপায়ে Object কে Initialize করা যায়?

উত্তরঃ ৩ টি উপায়ে অবজেক্ট ইনিশিয়ালাইজ করা যায় :

- i. রেফারেন্স তেরিয়াবল্ দ্বারা



```
class A{
}
class B{
}
class C: public A, public B{
}
class D: public A{
}
class X: public D, public C{
}
```

প্রশ্ন ১০. জাভাতে কেন মাল্টিপ্ল ইনহেরিটেন্স সার্পোট করে না?

উত্তরঃ মাল্টিপ্ল ইনহেরিটেন্সের ক্ষেত্রে একটি বড় সমস্যা হলো ডায়মন্ড সমস্যা (Diamond Problem)। একটি উদাহরণ দেওয়া যাক, মনে করা যাক, একটি ক্লাস A, একে B ও C এক্সেন্ট করে। আবার আরেকটি ক্লাস D যা কিনা B ও C কে এক্সেন্ট করে। এখন যদি A এর কোনো একটি মেথডকে B ও C দুটিই উভাররাইট করে এবং D থেকে সেই মেথডটি কল করা হয়, তাহলে সেটি কোন ক্লাসের মেথডকে কল করবে? B না C? সমস্যাটি নিচ্ছি বোঝা যাচ্ছে। জাভা ডিজাইনাররা এরকম সমস্যার মধ্য দিয়ে যেতে চায়নি। তাই জাভাতে মাল্টিপ্ল ইনহেরিটেন্স নেই।

তাই জাভা ডিজাইনাররা জটিলতা ক্রান্ত করতে এবং ভাষা সরল করতে মাল্টিপ্ল ইনহেরিটেন্স জাভাতে সমর্থন করেনি।

```
class A{
    void msg(){System.out.println("Hello");}
}
```

```
class B{
    void msg(){System.out.println("Welcome");}
}
```

```
class C extends A,B{//suppose if it were
```

```
public static void main(String args[]){
    C obj=new C();
    obj.msg(); //Now which msg() method would be
    invoked?
}
```

- ii. মেথড দ্বারা
- iii. কনস্ট্রাক্টর দ্বারা

১) রেফারেন্স তেরিয়াবল্ দ্বারা

```
class Student{
    int id;
    String name;
}
class TestStudent2{
    public static void main(String args[]){
        Student s1=new Student();
        s1.id=101;
        s1.name="rkb";
        System.out.println(s1.id+" "+s1.name); //printing
        members with a white space
    }
}
```

২) মেথড দ্বারা

```
class Student{
    int rollno;
    String name;
    void insertRecord(int r, String n){
        rollno=r;
        name=n;
    }
    void
    displayInformation(){System.out.println(ro
    llno+" "+name);}
}
class TestStudent4{
    public static void main(String args[]){
        Student s1=new Student();
        Student s2=new Student();
        s1.insertRecord(111,"RKB");
        s2.insertRecord(222,"SADIK");
        s1.displayInformation();
        s2.displayInformation();
    }
}
```

Output:

111 RKB

222 SADIK

[Islami Bank Officer (software engineer)-2019]

৩) কনস্ট্রাক্টর দ্বারা একটি ক্লাসের ইনস্ট্যান্স তৈরি করার সময় কনস্ট্রাক্টর মেথড বহুবিকল্প কল হয়। কনস্ট্রাক্টর মেথড মূলত ব্যবহার করা হয় ক্লাসের ভ্যারিয়েবলগুলোর প্রাথমিক মান নির্ধারণ করা জন্য এবং ক্লাসটিকে কাজ করতে প্রস্তুত করা জন্য। কনস্ট্রাক্টর মেথডের নাম আর ক্লাসের নাম একই হতে হয় এবং মেথডটি প্যারামিটার এবং করলেও কোনো রিটার্ন টাইপ থাকেনা। কোনো কনস্ট্রাক্টর মেথড ডিক্লেয়ার করা না হলে রিটার্ন টাইপ থাকেনা।

ডিফল্ট কনস্ট্রাক্টরকে কল করা হয়। ওভারলোডিং এর মাধ্যমে একটি ক্লাসে একাধিক কনস্ট্রাক্টর ডিক্লেয়ার করা যায়।

প্রশ্ন ১৩. Constructor কি?

উত্তরঃ কনস্ট্রাক্টর হল একটি ক্লাসের একটি বিশেষ ধরনের member function। অবজেক্ট initialization করলে সাথে সাথে ওই ক্লাস এর একটা default constructor auto create হয়। অবজেক্ট initialization করলে সাথে সাথে ওই ক্লাস এর একটা default constructor auto create হয়।

- a. constructor এর ভিত্তির return থাকে না এবং constructor এর নাম ক্লাস এর নামের অনুজ্ঞা হয়।
- b. default constructor ক্লাস এর ভিত্তির must থাকবে।
- c. Constructor সবসময় public mode এ declar করতে হয়।

উদাহরণ

```
#include<iostream>
using namespace std;
class Example {
    int a, b;
    Example() {
        a = 10;
        b = 20;
    }
    void Display() {
        cout << "Values :" << a << "\t" <<
        b;
    }
    int main() {
        Example Object;
        Object.Display();
        return 0;
    }
}
```

বিশেষ প্রটো্ট্যু:

মেথড এবং constructor এর মধ্যে main deference হলো মেথড এর রিটার্ন টাইপ থাকে এবং constructor এর রিটার্ন টাইপ থাকে না।

প্রশ্ন ১৪. Constructor কি কোন ভ্যালু রিটার্ন করে?

উত্তরঃ যদি constructor implicitly class এর current instance কে রিটার্ন করে। কিন্তু explicitly constructor এ return type ব্যবহার করা যায় না।

প্রশ্ন ১৫. Constructor কি ওভারলোড করা যায়??

উত্তরঃ যদি constructor ফাংশন এর আর্গুমেন্টের টাইপ অথবা আর্গুমেন্টের সংখ্যা পরিবর্তন করে ওভারলোড করা যায়।

প্রশ্ন ১৬. Types of Constructors

Constructors are three types :

- a. Default Constructor

b. Parametrized Constructor: যদি কোন constructor এর parameter থাকে, তাকে Parametrized Constructor বলে।

Example:

```
Class Student{
    int roll;
    Student (int r){
        this.roll = r;
        cout<<this.roll;
    }
}
int main(){
    Student obj(10);
    return 0;
}
```

c. Copy Constructor**প্রশ্ন ১৭. What is Copy Constructor?**

উত্তর: এটি একটি special type of Constructors যা একই class এর একটি object এর data অন্য object এ copy করে।

Syntax:

```
class class-name{
    Access Specifier:
    Member-Variables
    Member-Functions
    public:
        class-name(variable){
            // Constructor code
        }
        ... other Variables & Functions
}
```

Syntax : Argument In Main Function
ClassName Object1(value);
ClassName Object2=Object1;

প্রশ্ন ১৮. Destructor বলতে কী বোবার?

উত্তর: ক্লাসটির হচ্ছে এমন একটি ফাংশন যা কলস্ট্যাকের কর্তৃক ব্যবহৃত হোমোরিস্পেস খালস করে বা খালি করে দেয়।

Destructor বৈশিষ্ট্য:

- ক্লাস এর নাম ও destructor এর নাম একই হতে হবে। আগে একটি টিক (~) থাকে।
- destructor আর্গুমেন্ট নিতে পারে না।
- destructor এর কোন রিটার্ন টাইপ নেই।

Syntax:

```
class name{
    class name(){}
    ~class name();
}
```

Example:

```
#include<iostream>
using namespace std;
class Marks{
```

```
public:
    int maths;
    int science;
    Marks() { //constructor
        cout << "Inside Constructor" << endl;
        cout << "C++ Object created" << endl;
    }
    ~Marks() { //Destructor
        cout << "Inside Destructor" << endl;
        cout << "C++ Object destructed" << endl;
    }
}
```

Output:

```
Inside Constructor
C++ Object created
```

```
Inside Destructor
C++ Object destructed
```

প্রশ্ন ১৯. Why are constructor and destructor functions used in object oriented programming?

Give examples of each function in C++ or java language. [Ministry of Agriculture - SAE- 2021]

উত্তর: কনস্ট্যাক্টর মেমোরি ব্যবহৃত করতে এবং ক্লাসের অবজেক্ট তৈরি করতে প্রয়োজন হয়, যেখানে, একটি অবজেক্ট খালস করতে বা ব্যবহৃত হোমোরিস্পেস খালি করতে destructor প্রয়োজন হয়। তাই অবজেক্ট এরিয়েটেড প্রয়োজিন এবং কলস্ট্যাকের ব্যবহার হয়।

উত্তর: উপরে দেওয়া আছে।

প্রশ্ন ২০. Constructors and methods এর মধ্যে পার্শ্ব্যত্বে লিখ?**উত্তর:**

Constructor	Method
কোন অবজেক্টের state initialize করতে কলস্ট্যাক্টর ব্যবহার করা হয়।	মেথডের সাহায্যে অবজেক্টের behavior প্রকাশ করা হয়।
রিটার্ন টাইপ থাকে না	রিটার্ন টাইপ থাকে
কলস্ট্যাকে সরাসরি কল করা যায় না। নতুন একটা অবজেক্ট তৈরি হলে এটি কল হয়।	মেথডকে সরাসরি কল করা যায় না। সেই সব অবজেক্ট এর মাধ্যমে যাদেরকে তৈরি করা হয়েছে।
আভা কলস্ট্যাকের একটি ডিফল্ট কলস্ট্যাকের সরবরাহ করে যদি আপনার ক্লাসে কোনও কলস্ট্যাকের না থাকে।	আভা কলস্ট্যাকের ডিফল্ট method সরবরাহ করে।
কলস্ট্যাকের এর নাম অবশ্যই ক্লাস এর নাম এর নামে হতে হবে। তারা কিছুই রিটার্ন করতে পারে না।	মেথড অবশ্যই কিছু রিটার্ন করতে হবে যদিও এটি void হতে পারে।

প্রশ্ন ২১. পলিমোরফিজম কাকে বলে? [NTRCA-2014, Bangladesh Bank - 2014, PSC-2017].

উত্তর: পলিমোরফিজম শব্দটি 'পলি' এবং 'ম্যারফিজম' নামক দুটি শীর্ষ শব্দের সমন্বয়ে গঠিত। যদের অর্থ যথাক্রমে বহু এবং ক্লাস। অর্থাৎ ভিন্ন ভিন্ন অবস্থায় একই অবজেক্টের বহুরূপ ধারণ করা বা তিনি তিনি প্রদান করার প্রক্রিয়াকে পলিমোরফিজম বলে।

```
};  
int main() {  
    Shape *shape;  
    Rectangle rec(10,7);  
    Triangle tri(10,5);  
    shape = &rec;  
    shape->area();  
    shape = &tri;  
    shape->area();  
    return 0;  
}
```

প্রশ্ন ২২. যাসেজ পাসিং বলতে কী বোবার?

উত্তর: যে প্রক্রিয়ায় অবজেক্টসমূহ পরপরের প্রয়োজনে একে অপরের কাছে অনুরোধ পাঠায় এবং প্রেরিত অনুরোধের প্রেক্ষিতে ফলাফল প্রেরণ করে নিজেদের মধ্যে সংযোগ তৈরী করে, তাকে যাসেজ পাসিং বলে।

প্রশ্ন ২৩. Compile-time polymorphism and runtime polymorphism এর মধ্যে পার্শ্ব্যত্ব কোন লিখ?**উত্তর:**

Compile-Time Polymorphism	Runtime polymorphism
compile-time polymorphism এ কোন মেথড কল হয়েছে সেটি কলস্ট্যাক টাইমে সিদ্ধান্ত হয়।	runtime polymorphism, এ কোন উভারলিভেন মেথড কল হয়েছে সেটি runtime টাইমে সিদ্ধান্ত হয়।
এটিকে static binding, early binding, or overloading বলা হয়।	এটিকে dynamic binding, late binding, overriding, or dynamic method dispatch বলা হয়।
কলস্ট্যাক টাইম পলিমোরফিজম অর্জনের জন্য উভারলেডিং ব্যবহার করা।	runtime পলিমোরফিজম অর্জনের জন্য Overriding ব্যবহার করা।
fast execution	slower execution কারণ উভারলিভেন মেথড কল হয়েছে সেটি runtime টাইমে সিদ্ধান্ত হয়।
less flexibility কারণ সবকিছুই compile টাইমে সিদ্ধান্ত হয়।	more flexibility কারণ সবকিছুই runtime টাইমে সিদ্ধান্ত হয়।

প্রশ্ন ২৪. Runtime Polymorphism কি?

উত্তর: গ্লান্টাইম পলিমোরফিজম বা ডায়ানামিক মেথড dispatch এমন একটি প্রক্রিয়া যাধ্যমে কোন উভারলিভেড মেথডটি কল হয়েছে সেটি কলস্ট্যাক টাইমে না করে গ্লান্টাইমে সমাধান করা হয়। এই প্রক্রিয়াতে, একটি সুপারক্লাসের রেফারেন্সের মাধ্যমে উভারলিভেড মেথড কল করা হয়। কোন মেথডটি কল হবে সেটি রেফারেন্স ভেরিয়েবল থাকা উল্লেখ করা অবজেক্টের উপর ভিত্তি করে ঠিক করা হয়।

```
class Bike{  
    void run(){System.out.println("running");}  
}
```

```
class Splendor extends Bike{
    void run(){
        System.out.println("running safely with 60km");
    }
}
```

```
public static void main(String args[]){
    Bike b = new Splendor(); //upcasting
    b.run();
}
```

Output: running safely with 60km.

In this process, an overridden method is called through the reference variable of a superclass. The determination of the method to be called is based on the object being referred to by the reference variable.

প্রশ্ন ২৫. Encapsulation কাকে বলে? বা ভাটা হাইটি কী?

উত্তরঃ Encapsulation এর অপর নাম Data hiding বা Data protection। যে প্রক্রিয়া ভাটা ও ফাংশন সমূহকে একত্রিত করে ক্লস টৈকী করা যায় এবং ভাটা ও ফাংশন সমূহকে বাইরের অবরুদ্ধত এক্সেস হতে রক্ষা করা যায় তাকে এনক্যাপ্সুলেশন বলে।

```
#include<iostream>
class sum{
    private: int a,b,c;
    public: void add(){
        cout<<"Enter any two numbers: ";
        cin>>a>>b;
        c=a+b;
        cout<<"Sum: "<<c;
    }
};

int main(){
    sum s;
    s.add();
    return 0;
}
```

প্রশ্ন ২৬. ভাটা আ্যাবট্রাকশন কাকে বলে?

উত্তরঃ ভাটা আ্যাবট্রাকশন হচ্ছে এমন এক ধরনের ভাটা টাইপ বৰ্ণনা কৌশল যেখানে কিছু ভাটা থাকে এবং এই ভাটা এক্সেস করার জন্য কতভাবে ফাংশন থাকে।

```
#include<iostream>
class sum{
    private: int a,b,c;
    public: void add(){
        cout<<"Enter any two numbers: ";
        cin>>a>>b;
        c=a+b;
        cout<<"Sum: "<<c;
    }
};

int main(){
    sum s;
    s.add();
    return 0;
}
```

C তে লেখা সকল প্রোগ্রাম C++ এ লিখতে চাইলে শুধুমাত্র C++ এর #include<iostream> এই ভেতর টি এখনে লিখতে হবে এবং using namespace std এই Name Space টি ব্যবহার করতে হবে। C++ এ C এর অন্য সকল ফাংশন, Key-Word Support করে। C তে scanf() & printf() ফাংশন ব্যবহার করা হয় ইনপুট এবং আউটপুট নেওয়ার জন্য আর C++ এর cin & cout ফাংশন ব্যবহার করা হয় ইনপুট এবং আউটপুট নেওয়ার জন্য নিচে কিছু সাধারণ প্রোগ্রাম এর উদাহরণ দেয়া হল C++ এ।

■ দিনকে বছরে বা মাসে পরিবর্তনের সি ++ ব্যবহার করে প্রোগ্রাম।

```
#include<iostream>
using namespace std;
int main(){
    int days,years,weeks,num1;
    cout<<"Enter the number of days : "<< endl;
    cin>>days;
    years=days/365;
    weeks=days/7;
    cout << years << " years or ";
    cout << weeks << " weeks ";
    return 0;
}
```

■ আজকের তারিখ দেখার জন্য সি ++ ব্যবহার করে প্রোগ্রাম।

```
#include<iostream>
#include<ctime>
using namespace std;
int main(){
    time_t now = time(0);
    struct tm *lmt = localtime(&now);
    cout<<"Current system date is
"<<lmt->tm_mday<<"/"<<1 + lmt-
>tm_mon<<"/"<<1900 + lmt->tm_year;
    return 0;
}
```

Output: Current system date is 27/12/2020

1. a) নির্ণয়িত সিরিজের ঘোফলের জন্য একটি ভাটা / সি ++
গ্রেডার লিখুন? $\frac{1}{1!} + \frac{2}{2!} + \frac{3}{3!} + \dots + \frac{N}{N!}$

```
#include<iostream>
using namespace std;
double seriesSum(int num){
    double res = 0, fact = 1;
    for (int i = 1; i <= num; i++){
        fact = fact * i;
        res = res + (i / fact);
    }
    return(res);
}

int main(){
    int n;
    cout<<"Enter Input n: ";
    cin>>n;
    cout << "Sum: " << seriesSum (n);
    return 0;
}
```

১. JAVA

প্রশ্ন ১. জাভা কি? [what is java]

উত্তরঃ জাভা একটি প্রোগ্রামিং ভাষা। Java কে fully object oriented programming language বলা হয়। সান মাইক্রোসিস্টেম ১০ এর দশকের গোড়ার দিকে জাভা ডিজাইন করার পরে এটি অতি দ্রুত বিশ্বের সবচেয়ে জনপ্রিয় প্রোগ্রামিং ভাষার একটিতে পরিণত হয়। জাভার এই জনপ্রিয়তার মূল কারণ এর বহনযোগ্যতা (portability), নিরাপত্তা, এবং অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং ও ওব্রেব প্রোগ্রামিং এর প্রতি পরিপূর্ণ সাপোর্ট।

প্রশ্ন ২. জাভাকে কেন প্রাটফর্ম ইডিপেন্ডেন্ট প্রোগ্রামিং ভাষা বলা হয়? [why Java is called platform independent?] [BPSC Assistant Programmer 2019]

উত্তরঃ জাভা কোড কোন মেশিনকে উদ্দেশ্য না করে শুধুমাত্র JVM মেশিনকে উদ্দেশ্য করে লিখা হয়। আব জেভিএম অনেকগুলো মেশিনের জন্য লিখা হয়েছে যেমন-উইন্ডোজ, লিনাক্স, ম্যাক গ্যাস ইত্যাদি। যেহেতু সব মেশিনের জন্যই জেভিএম রয়েছে তাই আমাদের তৈরিকৃত কোড যেকোন মেশিনেই চলবে। তাই এটাকে মেশিন ইডিপেন্ডেন্ট বলা হয়।

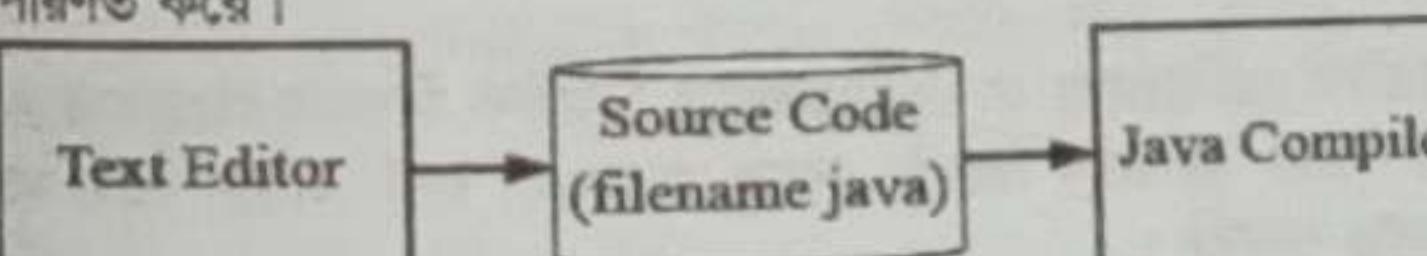
প্রশ্ন ৩. Java language এর Feature কোন লিখ? [Write down the feature of Java language]

উত্তরঃ

- Simple
- Object-Oriented
- Portable
- Platform Independent
- Secured
- Robust
- Architecture Neutral Interpreted
- High Performance
- Multithreaded
- Distributed
- Dynamic

প্রশ্ন ৪. জাভা কিভাবে কাজ করে?

উত্তরঃ জাভাতে লেখা কোড প্রথমে কম্পাইল হয়ে বাইটকোডে (Bytecode) রূপান্তরিত হয়। এই বাইটকোড পরবর্তী জাভা ভার্চুয়াল মেশিনের JIT কম্পাইলারের মাধ্যমে ইন্টারপ্রিট করে মেশিন ল্যাঙ্গুয়েজে পরিণত করে।



The process to compile a java program...

প্রথমে জাভা কোডকে কম্পাইল করা হয় এবং পরে আউটপুটকে ইন্টারপ্রিট করা হয় বলে জাভাকে একই সঙ্গে কম্পাইল এবং ইন্টারপ্রিট ল্যাঙ্গুয়েজ বলা হয়।

প্রশ্ন ৫. বাইটকোড কি? [What is bytecode]

উত্তরঃ জাভা কোড লিখার পর জাভা কম্পাইলার দ্বাৰা প্রস্তুতকৃত কোড যা JVM মেশিনের জন্য উপযোগী। অর্থাৎ ডিট্যুম্যান ল্যাঙ্গুয়েজ এবং মেশিন ল্যাঙ্গুয়েজের মধ্যবর্তী কোডই বাইটকোড।

প্রশ্ন ৬. What gives Java its 'write once and run anywhere' nature?

উত্তরঃ জাভা বাইটকোড জাভা compiler জাভা প্রোগ্রামকে একটি ড্রাইভ কাইল (বাইট কোড) রূপান্তর করে যা source কোড এবং মেশিন কোডের মধ্যে intermediate ভাষা। এই বাইটকোড প্র্যাটফর্ম নিন্তি নয় এবং যে কোণও কম্পিউটারে চালানো যেতে পারে।

প্রশ্ন ৭. আইডিই(IDE) ব্যবহারের সুবিধাগুলো কি?

উত্তরঃ IDE এর মূল উদ্দেশ্য হলো কোডিং এর সহজ বাচানো। নিচে এর কিছু সুবিধা উল্লেখ করা হলো-

১. স্টার্ট কোড ইডিটর থাকে যা বানান ভুল সংশোধন করে, বিভিন্ন কিওর্ড, মেথডকে ইন্টালাইট করার হাইলাইটার টুলস আছে, কোড জেনারেশন সুবিধা ইত্যাদি।
২. সহজেই কোড ডিবাগ করা যায়।
৩. এক ক্লিকেই কোড বান ও কম্পাইল করে আউটপুট দেখা যায়।
৪. প্রোগ্রামের আউটপুট একই স্থানে কোডের মান উন্মুক্ত সহজভাবে করে।

প্রশ্ন ৮. JDK, JRE এর মধ্যে পার্থক্য কি? [Islami Bank Officer(software engineer)-2019]

উত্তরঃ JRE: জাভা প্রোগ্রামকে রান করতে কম্পকে যতটুকু টুলস লাগে সেগুলোর একটা প্যাকেজ হলো Java Runtime Environment বা JRE।

JDK: জাভা ডেভেলপমেন্ট কিট হচ্ছে JRE এবং জাভা কোড লিখতে যতধৰণের টুলস লাগে তার একটি পরিপূর্ণ প্যাকেজ। অর্থাৎ জাভা প্রোগ্রাম করার জন্য শুধুমাত্র JDK হলো চলবে কারণ এতে জাভা কম্পাইলার, জাভা ডিবাগার, ডকুমেন্টেশন জেনারেটর ইত্যাদি থাকে।

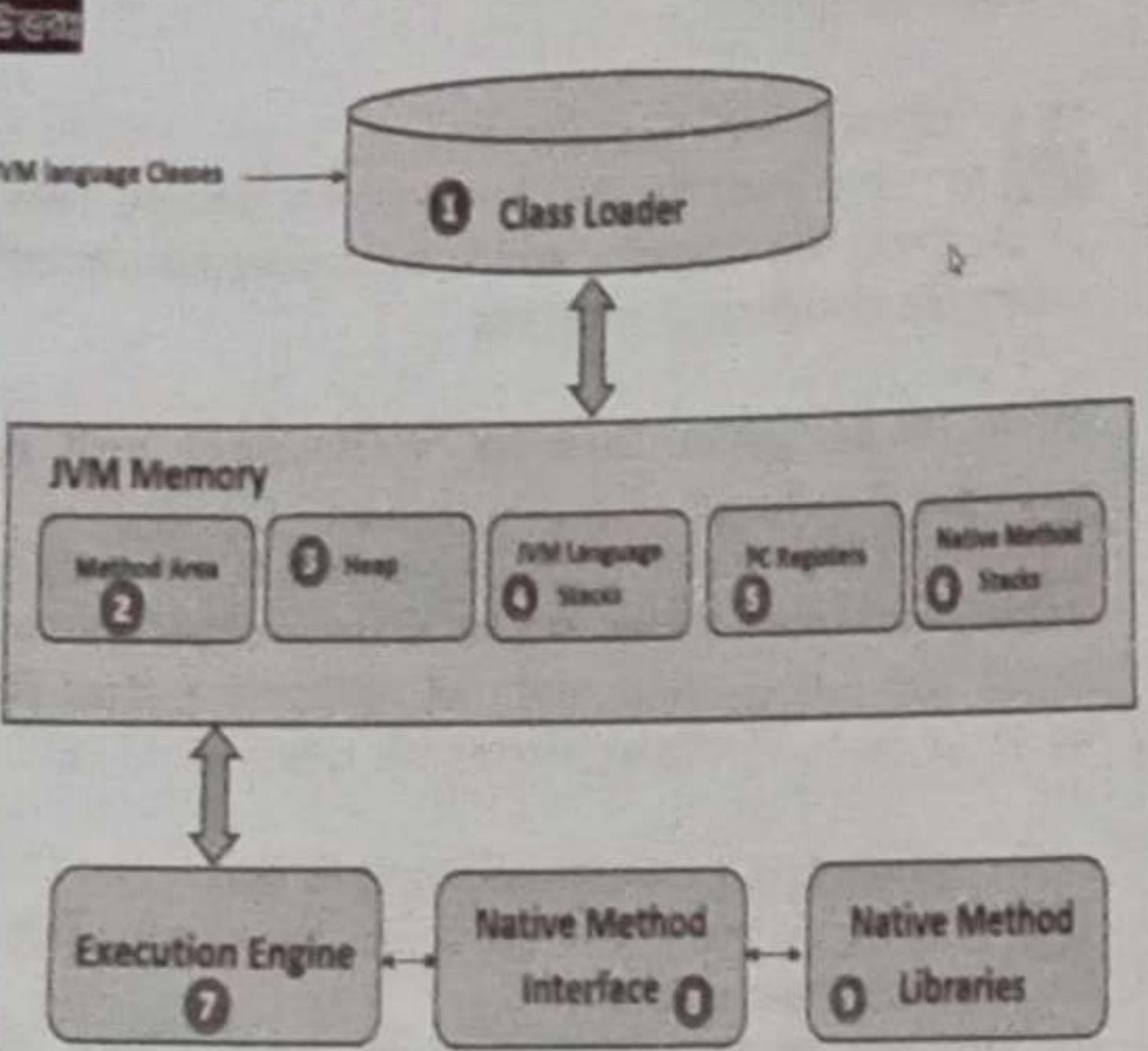
প্রশ্ন ৯. Java virtual machine(JVM) কি? [Islami Bank Officer(software engineer)-2019] What is virtual machine? [প্লিটেকেনিক ইনসিটিউট ইনস্ট্রাউন্স- ২০০৫]

উত্তরঃ ভার্চুয়াল মেশিন হলো একটি সফ্টওয়ার। যা একটি কম্পিউটার এর ভিতর আর একটি ভার্চুয়াল কম্পিউটার এর মতো কাজ করে। যেমন জাভা ভার্চুয়াল মেশিন সান মাইক্রোসিস্টেম-এর একটি সফ্টওয়ার যা বাইটকোড বান করতে সক্ষম। এটি জাভা বাইটকোডের একটি প্রোগ্রামকে মেশিন ভাষায় পরিবর্তিত হয়ে তা রান করে। এটি জাভা সফ্টওয়্যার প্লাটফর্মের কোড একজিকিউশনের সহজযোগ্য একটি উপাদান।

জাভা কম্পাইলার দিয়ে কম্পাইল করা কোডগুলো মূলত বাইটকোডে (Bytecode) রূপান্তরিত হয়। এটি জাভা বাইটকোডের (অর্থাৎ ভাষা) একটি প্রোগ্রামকে মেশিন ভাষায় পরিবর্তিত ও তা রান করে। এটি জাভা প্রোগ্রামকে দোষ platform independence এবং সব platform তথ্য সব অপারেটিং সিস্টেমে চলার ability রয়েছে।

Object Oriented Programming

প্রশ্ন ১০. JVM দ্বাৰা কতগুলো মেমৰি এৰিয়া allocated কৰা হয়।
Islam Bank Officer(software engineer)-2019]



Many types:

১. ক্লাস লোডারঃ এটি জেভিএম এর একটি সাব সিস্টেম যা ক্লাস ফাইলকে লোড করে।
২. ক্লাস (মেথড) এরিয়াঃ এটা প্রতিটি ক্লাসের স্ট্যাকচারকে সংরক্ষণ করে রান্টাইম কনস্ট্যান্ট পুল, ক্লিন এবং মেথড ডাটা, মেথড ডাটার কোড হিসেবে।
৩. হিপ্প এটা হচ্ছে রান্টাইম ডাটা এৰিয়া যেখানে অজেন্টে এর জন্য মেমৰি এলাকেট কৰা হয়।
৪. স্ট্যাক জাভার স্ট্যাক ফ্রেমকে সংরক্ষণ করে। এটা লোকাল ভেরিয়েবল ও সাময়িক রেজাল্টকে ধৰণ কৰে মেথড কে কলং এবং রিটৰ্ন এর যত ডাটা আছে সবকিছু ধৰণ কৰে।
৫. প্ৰোগ্ৰাম কাউন্টাৰ (pc): এটা এক ধৰণেৰ রেজিস্টাৱ। বৰ্তমানে যেসব ইন্ট্ৰুকশন জাভা ভাৰ্যাল মেশিনে এক্সেকিউট হচ্ছে তাদেৱ এড্ৰেস ধৰণ কৰাই এৰ কাজ।
৬. মেথড মেথড স্ট্যাকঃ এটা একটি এপ্লিকেশনে যে সকল মেথড সাধাৰণত ব্যবহৃত হয় সেই সকলগুলো মেথড ধৰণ কৰে।
৭. জাভা এক্সেকিউশন ইঞ্জিন একটি এক্সেকিউশন ইঞ্জিনে ৩টি সাব কম্পোনেন্ট রয়েছেঃ
 - i. একটি ভাৰ্যাল প্ৰসেসৰ
 - ii. ইন্টাৱিগ্রেটাৱ
 - iii. জাস্ট-ইন-টাইম (jit) কম্পাইলাৱ

প্রশ্ন ১১. JIT কম্পাইলাৱ কি? এটা কিভাৱে কাজ কৰে?

উত্তৰ: জাস্ট-ইন-টাইম (JIT) কম্পাইলাৱ হচ্ছে JVM এৰ অংশ যা বাইট কোডকে প্ৰসেস কৰে থাকে। এটা গান্টাইমে অৰ্ধাং প্ৰসেস চলাকালীন সময়ে বাইটকোডকে ইন্টাৱিগ্রেট কৰে মেশিন কোডে কুণ্ডলিত কৰে। তাই এটাকে ডাইনামিক ট্ৰান্সলেটৱও বলা হয়।

প্রশ্ন ১২. What is package in java ?

উত্তৰঃ কত গুলো interrelated class এবং interface এৰ সংথাকে একত্ৰে বলা হয় package যা access protection এবং namespace management প্ৰদান কৰে।
Package সাধাৰণত ২ প্ৰকাৰ : যথাঃ

- i. Built in Package
 - ii. User define package
- Built in Package** সমূহৰ নাম নিচে দেয়া হলঃ
- (i) lang – Automatically imported and it contains language support classes.
 - (ii). io – Contains classes for input and output operations.
 - (iii). util – Contains utility classes for implementing data structures.
 - (iv). applet – This package contains classes that create applets.
 - (v). awt – Contain classes that implement compounds for GUI.
 - (vi). net – This package contains classes that support networking operations.

প্রশ্ন ১৩. জাভাতে প্যাকেজ ব্যবহাৰৰ সুবিধাগুলো লিখো?

- উত্তৰ:** জাভাতে প্যাকেজ ব্যবহাৰৰ সুবিধাগুলো:
- o Packages avoid the name clashes.
 - o The Package provides easier access control.
 - o We can also have the hidden classes that are not visible outside and used by the package.
 - o It is easier to locate the related classes.

প্রশ্ন ১৪. জাভাতে ভেরিয়েবল কত প্ৰকাৰ?

- উত্তৰ:** জাভাতে ভেরিয়েবল হচ্ছে তিন প্ৰকাৰ। এগুলো হলঃ
১. লোকাল ভেরিয়েবল
 ২. ইনস্ট্যাল ভেরিয়েবল
 ৩. স্ট্যাটিক ভেরিয়েবল

লোকাল ভেরিয়েবলটো মেধডেৱ ভিতৰে যেই ভেরিয়েবল ডিক্ৰেয়াৰ কৰা হয় তাকে লোকাল ভেরিয়েবল বলা হয়।

ইনস্ট্যাল ভেরিয়েবলটো যে ভেরিয়েবল ক্লাসেৰ মধ্যে ডিক্ৰেয়াৰ কৰা হয়, কিন্তু মেধডেৱ বাইৱে থাকে তাকে ইনস্ট্যাল ভেরিয়েবল বলা হয়। আৰু এটা স্ট্যাটিক এৰ মত কৰে ডিক্ৰেয়াৰ কৰা হয় না।

স্ট্যাটিক ভেরিয়েবলটো যে ভেরিয়েবলকে স্ট্যাটিক হিসেবে ডিক্ৰেয়াৰ কৰা হয়। এটা উদাহাৰণ দিলেই বুঝতে পাৰবেন। সেটা একটু পৰই হয়ত বুঝতে পাৰবেন।

প্রশ্ন ১৫. জাভাৰ ডাটা টাইপগুলোৰ ডিফল্ট ভেলু এবং সাইজ?

Data Type	Default Value	Default size
boolean	false	1 bit
char	'\u0000'	2 byte

Object Oriented Programming

byte	0	1 byte
short	0	2 byte
int	0	4 byte
long	0L	8 byte
float	0.0f	4 byte
double	0.0d	8 byte

```
int toNum=(int)letter;//type
casting explicitly from character to
integer type
System.out.println("Number : " +
toNum);// output 65.explicit casting
occured here
}
}
```

প্রশ্ন ১৭. Without main() method ছাড়া প্ৰোগ্ৰাম কি বাব কৰা যাবে?

উত্তৰ: হ্যা, মেইন মেথড ছাড়াই প্ৰোগ্ৰামটি চালানোৰ একটি উপায় হল স্ট্যাটিক ব্লক।

প্রশ্ন ১৮. What is MultiThreading in java ?

উত্তৰঃ একবোগে একাধিক শ্ৰেণি কাৰ্যকৰ কৰাৰ প্ৰক্ৰিয়া জাভাতে মাল্টিপ্ৰিডিং বলে। একটি শ্ৰেণি একটি sub-process, যা প্ৰক্ৰিয়াকৰণৰ মুদ্ৰণ ইউনিট। Multiprocessing এবং Multithreading উভয় Multitasking অৰ্জন কৰতে ব্যবহৃত হয়।

প্রশ্ন ১৯. What is Difference between process and thread ?

উত্তৰ:

Process	Thread
1.Program in Execution	1.Separate path of execution one or more threads is called as process
2.Heavy weight .	2.Light weight .
3.Requires separate address space.	3.shares same address space .
4.Inter process communication is expensive .	4.Inter thread communication is less expensive when compares to process .
5.Process can be divided into multiple threads.	5.Threads can not be subdivided.

প্রশ্ন ২২. সুপাৰ কীওয়াৰ্ড কি?

উত্তৰ: জাভাতে সুপাৰ কীওয়াৰ্ড হল একটি বেকারেজ ভেরিয়েবল যা immediate প্যারেন্ট ক্লাস অবজেক্টকে রেফাৰ কৰতে ব্যবহৃত হয়।

```
class Animal{
Animal(){}
System.out.println("animal is created");
}
class Dog extends Animal{
Dog(){}
System.out.println("dog is created");
}
}
class TestSuper4{
```

```
public static void main(String args[]){
    Dog d=new Dog();
}
```

প্রশ্ন 23. সুপার কীওয়ার্ড এর কাজ কি?

উত্তর: There are the following uses of super keyword.

- ০ সুপার কীওয়ার্ড হল ইমিডিয়েট প্যারেট ক্লাসের instance variable কে রেফার করতে ব্যবহৃত হয়।
- ০ immediate parent class এর মেথডকে কল করতে সুপার কীওয়ার্ড ব্যবহার করা হয়।
- ০ super() immediate parent class এর constructor কল করতে ব্যবহার করা হয়।

প্রশ্ন 24. This keyword and super এর মধ্যে পার্থক্য কীৰ্তি?

উত্তর:

- ০ super প্যারেট ক্লাসের context কে পর্যবেক্ষণ করতে ব্যবহার করা হয় যেখানে this keyword কারেন্ট ক্লাসের context কে পর্যবেক্ষণ করতে ব্যবহার করা হয়।
- ০ super কীওয়ার্ডটি প্রাথমিকভাবে ডিইভিউট ক্লাস কনস্ট্রাক্টরের মধ্যে বেস ক্লাস ভেরিয়েবলকে ইনিশিয়েলাইজ করার জন্য ব্যবহৃত হয় যেখানে ক্লাস কনস্ট্রাক্টরে পাস করার সময় this কীওয়ার্ডটি প্রাথমিকভাবে local এবং instance ভেরিয়েবলের মধ্যে পার্থক্য করতে ব্যবহৃত হয়।
- ০ সুপার এবং this কনস্ট্রাক্টরের ভিতরে অন্য স্টেটমেন্ট হতে হবে অন্যথায় কল্পাইলার একটি জটি প্রদর্শন করবে।

প্রশ্ন 25. নিচের প্রোগ্রামের আউটপুট কি হবে?

```
class Person {
    public Person() {
        System.out.println("Person class
constructor called");
    }
}

public class Employee extends Person {
    public Employee() {
        System.out.println("Employee class
constructor called");
    }
}

public static void main (String args[]){
    Employee e = new Employee();
}
```

Output: Person class constructor called
Employee class constructor called

Explanation:যদি কোন super() বা this() derived ক্লাস কনস্ট্রাক্টরের মধ্যে explicitly included করা না হয় তবে super() কল্পাইলার ঘরা implicitly invoked করা হয়। অতএব, এই ক্ষেত্রে, Person class কনস্ট্রাক্টরকে প্রথমে called হয় এবং তারপর Employee class constructor called হয়।

প্রশ্ন 26. Constructor এর ভিতরে this() and super() স্টেটমেন্ট কি ব্যবহার করা যায়?

উত্তর: না, কারণ this() and super() কনস্ট্রাক্টরের ভিতরে অন্য স্টেটমেন্ট হতে হবে।

Example:

```
public class Test{
    Test() {
        super();
        this();
        System.out.println("Test class object is
created");
    }
}

public static void main(String []args){
    Test t = new Test();
}
```

Output:

Test.java:5: error: call to this must be first statement in constructor

প্রশ্ন 27. Final, Finally and Finalize এদের মধ্যেকার পার্থক্যগুলো লিখুন?

উত্তর:

Final	Finally	Finalize
final keyword জাতির একটি reserve keyword। এটি কোন ভেরিয়েবল, মেথড ও ক্লাস কে অপরিবর্তন্য রাখার জন্য ব্যবহার করা হয়। কোন ভেরিয়েবল কে final ঘোষণা করলে JVM স্টেটিক ছুরু রান করে। আপনি যদি কোন final ঘোষণাকৃত কোন ভেরিয়েবল, মেথড ও ক্লাস মোডিফাই করতে যান তাহলে অবশ্যই compilation error হবে।	প্রোগ্রামের কোন ভর্তৃপূর্ণ কোড এক্সেকিউট করার জন্য এটি ব্যবহার করা হয়। কোন ভেরিয়েবল কে final ঘোষণা করলে JVM স্টেটিক ছুরু রান করে। আপনি যদি কোন final exception এক্সেপশন কে handle করেছি কি হয়নি যাই হোক না কেন এটি এক্সেকিউট হবে।	finalize() method টি object garbage collected হওয়ার ঠিক আগে কল করা হয়। এটি ক্লিনআপ প্রসেসিং সম্পাদন করতে ব্যবহৃত হয়।
Final is a keyword.	Finally is a block.	Finalize is a method.

প্রশ্ন 28. Java static methods এর restrictions কীলো কি কি?

উত্তর:

- ০ স্ট্যাটিক মেথড নন-স্ট্যাটিক ডেটা মেথার ব্যবহার করতে পারে না বা সরাসরি নন-স্ট্যাটিক মেথডকে কল করতে পারে না।
- ০ this এবং super static context এ ব্যবহার করা যাবে না কারণ এগুলি নন-স্ট্যাটিক।

প্রশ্ন 29. Java তে Main method static কেন?

উত্তর: কারণ স্ট্যাটিক মেথড কল করার জন্য অবজেক্টের প্রয়োজন নেই। যদি আমরা মেইন মেথড নন-স্ট্যাটিক করি তবে জেডিএমকে প্রথমে তার অবজেক্ট তৈরি করতে হবে এবং তারপরে main () মেথডটি কল করতে হবে যা অতিরিক্ত মেমরি ব্যবহৃতকে বাড়ায়।

প্রশ্ন 30. Why is method overloading not possible by changing the return type in java?

উত্তর: জাভাতে ambiguity এডানোর জন্য প্রোগ্রামের বিটার্ন টাইপ পরিবর্তন করে মেথড ওভারলেডিং করা যায় না।

```
class Adder{
    static int add(int a,int b){return a+b;}
    static double add(int a,int b){return a+b;}
}
```

```
class TestOverloading3{
    public static void main(String[] args){
        System.out.println(Adder.add(11,11));//ambiguity
    }
}
```

Output:

Compile Time Error: method add(int, int) is already defined in class Adder

প্রশ্ন 31. জাভায় Final keyword কি?

উত্তর: final keyword জাভার একটি reserve keyword। এটি কোন ভেরিয়েবল, মেথড ও ক্লাস কে অপরিবর্তন্য রাখার জন্য ব্যবহার করা হয়। কোন ভেরিয়েবল কে final ঘোষণা করলে JVM স্টেটিক ছুরু রান করে। আপনি যদি কোন final ঘোষণাকৃত কোন ভেরিয়েবল, মেথড ও ক্লাস মোডিফাই করতে যান তাহলে অবশ্যই compilation error হবে।

প্রশ্ন 32. Final variable কি?

উত্তর: যেকোন ধরনের ভেরিয়েবল(অথবা অবজেক্ট) যেমনঃ ক্লাস ভেরিয়েবল, ফিল্ড বা লোকাল ভেরিয়েবল (যেকোন ব্লক বা মেথডের মধ্যে) যদি final keyword ঘোষণা মোডিফাই করা হয়, তাহলে তাকে final ভেরিয়েবল বলে। জাভাতে, final ভেরিয়েবল ব্যবহারকারীরকে আপডেট করা থেকে সীমাবদ্ধ করতে ব্যবহৃত হয়। যদি আমরা final ভেরিয়েবল initialize করি তবে আমরা এর মান পরিবর্তন করতে পারে না।

```
class Bike9{
    final int speedlimit=90;//final variable
    void run(){
        speedlimit=400;
    }
}

public static void main(String args[]){
    Bike9 obj=new Bike9();
    obj.run();
}
//end of class
```

প্রশ্ন 33. Final method কি?

উত্তর: যদি কোন মেথড কে final keyword দ্বারা মোডিফাই করা হয়, তবে সে মেথড কে final মেথড বলে। আপনি যদি কোন মেথডের বৈশিষ্ট্য সার-ক্লাস দ্বারা পরিবর্তীত করতে না চান, তবে সে মেথডকে final keyword দ্বারা মোডিফাই করা যায়।

```
class Bike{
    final void run(){System.out.println("running");
}
```

```
class Honda extends Bike{
    void run(){System.out.println("running
safely with 100kmph");}
}
```

```
public static void main(String args[]){
    Honda honda= new Honda();
    honda.run();
}
}
```

Output:Compile Time Error

প্রশ্ন 34. Final method and abstract method এর মধ্যে পার্থক্য কি?

উত্তর: ফাইনাল মেথড এবং অবস্ট্রাক্ট মেথডের মধ্যে প্রধান পার্থক্য হল অবস্ট্রাক্ট মেথড ফাইনাল হতে পারে না কারণ এর ডেফলেটেন্স দেওয়ার জন্য আমাদের সাবক্লাসে সেগুলি ওভাররাইট করা দরকার।

প্রশ্ন 35. Final class কি?

উত্তর: যে সকল জাভা ক্লাস final keyword দ্বারা মোডিফাই করা হয়, সে সকল ক্লাস কে final ক্লাস বলে। String, Runtime এবং সকল র্যাপার ক্লাস final। আপনি এগুলো inherit বা modify করতে পারবেন না। আপনি যদি কোন ক্লাস কে ইনহেরিট করতে না চান, তবে ক্লাস টিকে final keyword দ্বারা মোডিফাই করতে পারেন। এতে আর সে ক্লাসকে ইনহেরিট করা যাবেন। তবে abstract class কে final করা যায় না।

```
final class Bike{}
class Hondal extends Bike{
    void run(){System.out.println("running safely with
100kmph");}
}
```

```
public static void main(String args[]){
    Hondal honda= new Hondal();
    honda.run();
}
}
```

Output:Compile Time Error

প্রশ্ন ৩৬. Final keyword এর সুবিধাগুলো কি?

উত্তর: final ভেরিয়েবল, মেষত ও ক্লাস দ্রুত কম্পাইল হয়। কারণ এভেন্যু অপরিবর্তনীয়। এটি দ্রুত সেফ বা সিংক্রনাইজ করার প্রয়োজন হয়না; কারণ ভেরিয়েবল কলস্টোরে মত এবং ক্লাস ইমিটেটরের হত আচরণ করে।

প্রশ্ন ৩৭. Final blank variable কি?

উত্তর: যে ফাইনাল ভেরিয়েবল, যোৰগুরু সময় ইনিশিয়ালাইজ হয় না তাকে ফাইনাল blank ভারিয়েবল বলে। আমরা সরাসরি ফাইনাল blank ভারিয়েবল ইনিশিয়ালাইজ করতে পারি না। ক্লাস কলস্টোর ব্যবহার করে আমাদের এটি ইনিশিয়ালাইজ করতে পারি।

```
class Student{
    int id;
    String name;
    final String PAN_CARD_NUMBER;
    ...
}
```

প্রশ্ন ৩৮. Final blank variable কি initialize করা যায়?

উত্তর: হ্যাঁ, এটি স্ট্যাটিক না হলে আমরা এটি constructor এ ইনিশিয়ালাইজ করতে পারি। যদি এটি স্ট্যাটিক ব্লক ফাইনাল ভারিয়েবল হয় তবে এটি কেবল স্ট্যাটিক ব্লকেই আরও করা যেতে পারে।

প্রশ্ন ৩৯. Main method কি final হিসেবে declare করা যায়?

উত্তর: হ্যাঁ, We can declare the main method as public static final void main(String[] args){}.

প্রশ্ন ৪০. নিচের প্রোগ্রামের আউটপুট বের করুন?

```
class Main {
    public static void main(String args[]){
        final int i;
        i = 20;
        System.out.println(i);
    }
}
```

Output: 20

Explanation: Since i is the blank final variable. It can be initialized only once. We have initialized it to 20. Therefore, 20 will be printed.

প্রশ্ন ৪১. নিচের প্রোগ্রামের আউটপুট বের করুন?

```
class Base {
    protected final void getInfo() {
        System.out.println("method of Base class");
    }
}

public class Derived extends Base {
    protected final void getInfo() {
        System.out.println("method of Derived class");
    }
}
```

```
} public static void main(String[] args) {
    Base obj = new Base();
    obj.getInfo();
}
```

Output:

```
Derived.java:11: error: getInfo() in Derived
cannot override getInfo() in Base
    protected final void getInfo()
                           ^
overridden method is final
1 error
```

Explanation:

The `getDetails()` method is final; therefore it can not be overridden in the subclass.

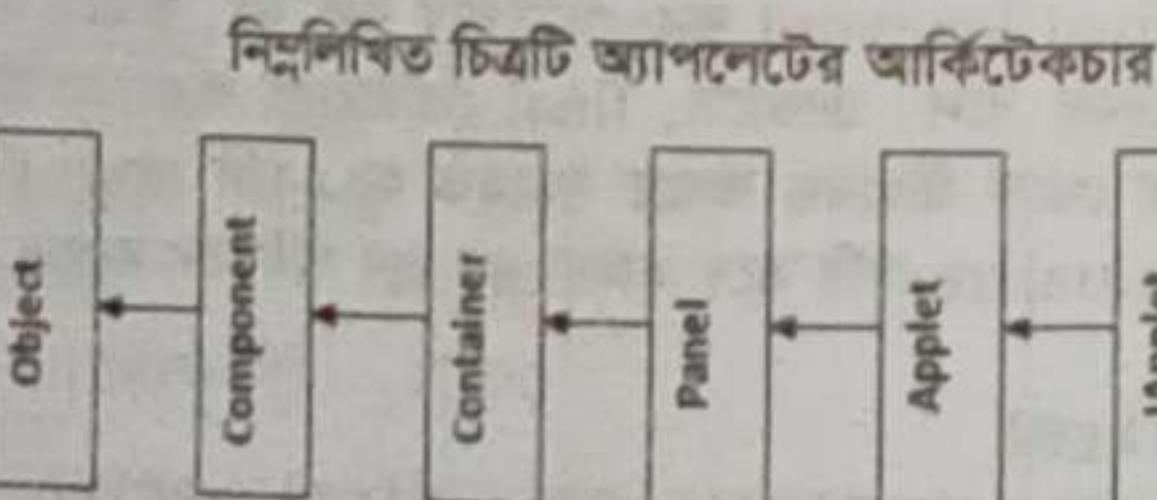
প্রশ্ন ৪২. জাতীয় InstanceOf operator কি?

উত্তর: জাতীয় instanceof টাইপ তুলনা অপারেটর হিসেবে পরিচিত কারণ এটি instance এর সাথে টাইপের তুলনা করে। এটি সত্য বা মিথ্যা return করে। যদি আমরা instanceof operator কেন ভেরিয়েবলের সাথে ব্যবহার করি, যাতে null value আছে তবে এটি মিথ্যা return করে। নিম্নলিখিত উদাহরণ বিবেচনা করুন।

```
class Simple1{
    public static void main(String args[]){
        Simple1 s=new Simple1();
        System.out.println(s instanceof
Simple1); //true
    }
}
```

প্রশ্ন ৪৩. Applet কি?

উত্তর: আপলেট একটি বিশেষ ধরনের প্রোগ্রাম যা webpage এর ভাষ্যামূলিক contain তৈরি করতে embedded করা হয়। এটি প্রাইজার ভিতরে রাখ হয় এবং ক্লায়েন্ট পাশে কাজ করে।



When an applet is created, the following methods are invoked in order.

- init()
- start()
- paint()

When an applet is destroyed, the following functions are invoked in order.

- stop()
- destroy()

প্রশ্ন ৪৪. Servlet কি?

উত্তর: Servlet একটি জাভা প্রোগ্রাম ভাষার ক্লাস যা সার্ভারগুলির ক্ষমতা প্রসারিত করার জন্য ব্যবহৃত হয়। যা request-response প্রোগ্রাম মডেলের মাধ্যমে আপ্লিকেশনগুলিকে হোস্ট করে।

-The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

-Finally: এর সংঘটিত হোক বা না হোক তারপরেও execute হতে হবে সেই code block কে finally block এ লেখা হয়।

প্রশ্ন ৪৫. Exception কি?

উত্তর: আমরা সাধারণত কোন প্রোগ্রাম লিখলে সেখানে কিছু তুল থাকতেই পারে যেটাকে আমরা Error হিসেবে জানি। Java এ একেই Exception বলে। সাধারণত প্রোগ্রাম এর কোথাও Exception ঘটলে প্রোগ্রাম টা বন্ধ হয়ে যায়, যদিও তার পরে আরও কোড থাকে। কিন্তু Java এ এই তুল গুলো হ্যান্ডেল করা যায় যার ফলে প্রোগ্রাম এর কোথাও ইরে ঘটলেও পুরু প্রোগ্রাম বন্ধ হয় না।

প্রশ্ন ৪৬. জাতীয়তে কত ধরনের Exception ঘটতে পারে?

উত্তর:

ধূই একার checked and unchecked। কিন্তু গোকলের মতে তিনি একার।

1. Checked Exception: Checked Exception গুলো কম্পাইল টাইমে checked হয়। যেমন, SQLException, ClassNotFoundException, etc.
2. Unchecked Exception: Unchecked Exception গুলো Run টাইমে checked হয়। যেমন, ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException, etc.
3. Error: এর হলে প্রোগ্রাম এক্সিট করে কারন এর রিকাভারেবল না। যেমন, OutOfMemoryError, AssertionError, etc.

প্রশ্ন ৪৭. Exception Handling কি?

উত্তর:

এমন একটি মেকানিজ্ম যার মাধ্যমে রানটাইম এর কে হ্যান্ডেল করা হয় এটি প্রাথমিকভাবে checked exception handle করতে ব্যবহৃত হয়। exception handling প্রোগ্রামের সাভাবিক ধৰাহ বজায় রাখে।

প্রশ্ন ৪৮. Checked Exception and Unchecked Exception এর মধ্যে পার্থক্য তুলো শিখ?

উত্তর:

1) Checked Exception

যে ক্লাস RuntimeException এবং Error ব্যতীত Throwable class কে extend করে তাকে checked exceptions বলে। e.g., IOException, SQLException, etc. Checked Exception গুলো কম্পাইল টাইমে checked হয়।

2) Unchecked Exception

যে ক্লাস RuntimeException কে extend করে তাকে unchecked exceptions বলে। e.g., ArithmeticException, NullPointerException, etc. Unchecked Exception গুলো Run টাইমে checked হয়।

প্রশ্ন: About Java try, catch and finally.

-The try statement allows you to define a block of code to be tested for errors while it is being executed.

-The catch statement allows you to define a block of code to be executed, if an error occurs in the try block.

-Finally: এর সংঘটিত হোক বা না হোক তারপরেও execute হতে হবে সেই code block কে finally block এ লেখা হয়।

প্রশ্ন ৪৯. প্রতিটি ট্রাই block এর পরে একটি ক্যাচ block থাকবে উভটির পক্ষে ব্যক্তি স্বত্ত্ব ব্যবহৃত করা হবে না? or Java try and catch Example.

উত্তর: এটি প্রয়োজন নেই যে প্রতিটি try block অবশ্যই একটি catch block অনুসরণ করবে। এটি একটি ক্যাচ block বা একটি finally block অনুসরণ করতে পারে। সূতরাং যেই exception thrown হওয়ার সম্ভাবনা রয়েছে তা ঘোষণা করা উচিত। Consider the following example.

```
import java.util.*;
import java.lang.*;
import java.io.*;
public class Main {
    public static void main(String[ ] args) {
        try {
            int [ ] myNumbers = {1, 2, 3};
            System.out.println(myNumbers[10]);
        } catch (Exception e) {
            System.out.println("Something
went wrong like:" + e);
        }
    }
}
```

প্রশ্ন ৫০. নিচের প্রোগ্রামের আউটপুট বের করুন?

```
public class ExceptionHandlingExample {
    public static void main(String args[ ]) {
        try {
            int a = 1/0;
            System.out.println("a = "+a);
        } catch(Exception
e){System.out.println(e);}
        catch(ArithmeticException
ex){System.out.println(ex);}
    }
}
```

Output:

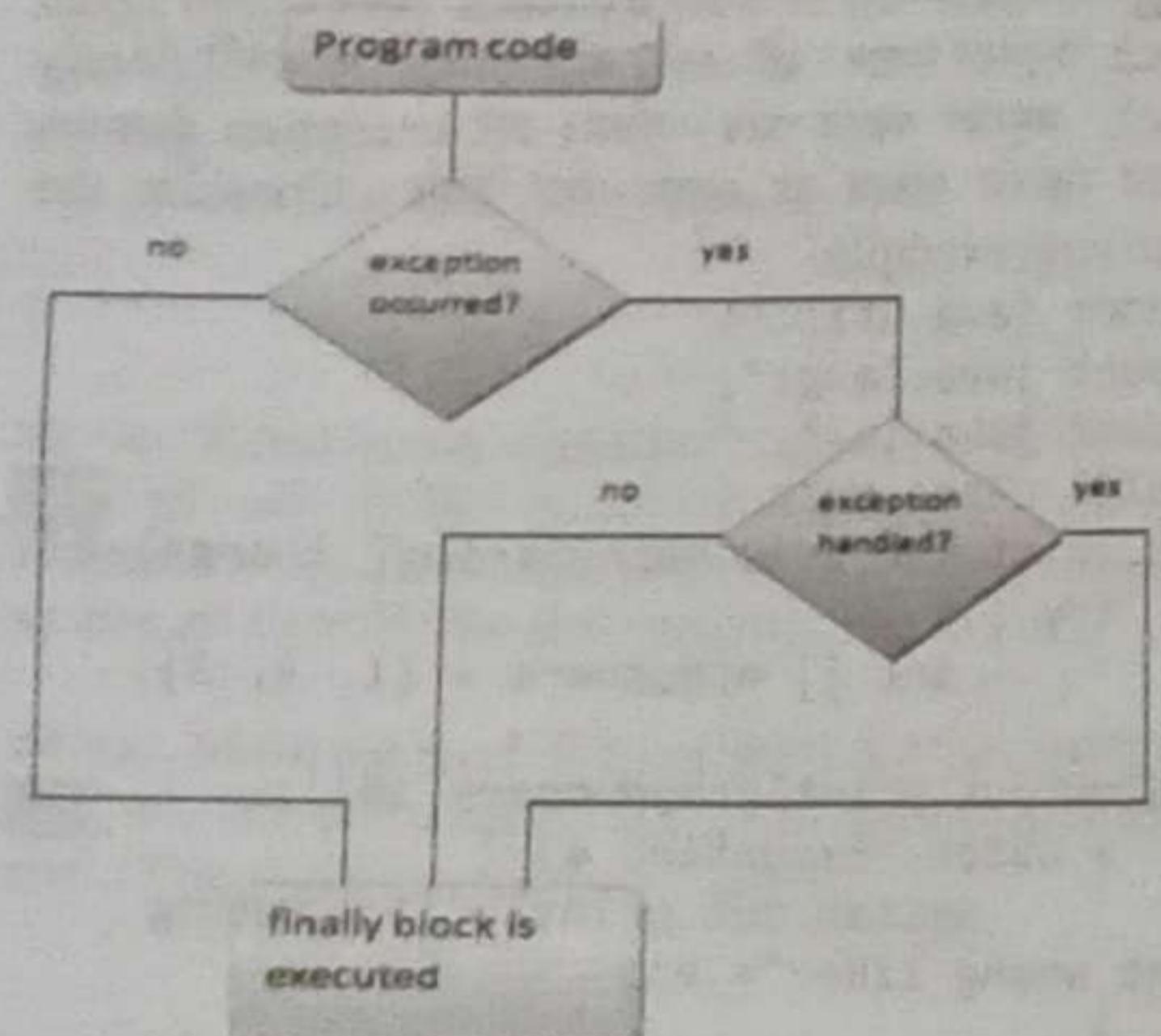
```
ExceptionHandlingExample.java:10: error: exception
ArithmeticException has already been caught
    catch(ArithmeticException
ex){System.out.println(ex);}
}
1 error
```

Object Oriented Programming

Explanation: ArithmeticException হল Exception এর সাবক্লাস। অতএব, এটি Exception এর পরে ব্যবহার করা যাবে না। যেহেতু Exception হল সমস্ত Exception এর জন্য বেস ক্লাস, তাই, Exception টি পরিচালনা করতে শেষ পর্যন্ত এটি ব্যবহার করা আবশ্যিক। এর পর কোন ক্লাস ব্যবহার করা যাবে না।

প্রশ্ন ১১. Finally block কি?

উত্তর: প্রয়োজনের কোন ক্লাসগুরু কোভ এক্সেকিউট করার জন্য এটি ব্যবহার করা হয়। exception handle হোলি কি হয়নি যাই যেকে না কেন এটি এক্সেকিউট হবে।



final Example:

```

public class Main {
    public static void main(String[] args) {
        try {
            int[] myNumbers = {1, 2, 3};
            System.out.println(myNumbers[10]);
        } catch (Exception e) {
            System.out.println("Something went wrong like:" + e);
        } finally {
            System.out.println("The 'try catch' is finished.");
        }
    }
}
  
```

প্রশ্ন ১২. Throw and Throws এর মধ্যে পার্থক্য কোন শিখ?

উত্তর:

Throw Keyword	Throws Keyword
1) The throw keyword is used to throw an exception explicitly.	The throws keyword is used to declare an exception.
2) The checked exceptions cannot be propagated with	The checked exception can be propagated with

propagated with throw only.	throws
3) The throw keyword is followed by an instance.	The throws keyword is followed by class.
4) The throw keyword is used within the method signature.	The throws keyword 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.

The throw keyword: The throw statement allows you to create a custom error. (নিজের মতো করে এর বানাতে throw ব্যবহার করা হয়)

Example:

```

public class Main {
    static void checkAge(int age) {
        if (age < 18) {
            throw new ArithmeticException("Access denied - You must be at least 18 years old.");
        } else {
            System.out.println("Access granted - You are old enough!");
        }
    }
    public static void main(String[] args) {
        checkAge(15); // Set age to 15 (which is below 18...)
    }
}
  
```

প্রশ্ন ১৩. কত উপরে জাভার Object তৈরি করা যায় ?

উত্তর: বিভিন্ন উপরে জাভার অবজেক্ট তৈরী করা যায় :

1. By new keyword
2. By newInstance() method
3. By clone() method
4. By deserialization
5. By factory method etc.

প্রশ্ন ১৪. নিচের প্রয়োজনের আউটপুট কোনটি?

উত্তর:

```

public class Main{
    public static void main(String []args){
        try {
            throw 90;
        }
    }
}
  
```

```

        }
        catch(int e){
            System.out.println("Caught the exception "+e);
        }
    }
}
  
```

Output:

Main.java:6: error: incompatible types: int cannot be converted to Throwable

 throw 90;

 ^

Main.java:8: error: unexpected type

 catch(int e){

 ^

 required: class

2 errors

Explanation: জাভাতে, throwable objects শব্দাত্মক করা যেতে পারে। যদি আমরা একটি integer object throw করার চেষ্টা করি, কম্পাইলার একটি error দেখাবে কারণ আমরা কোডের একটি ব্রক থেকে basic data type throw করতে পারি না।

প্রশ্ন ১৫. নিচের প্রয়োজনের আউটপুট কোনটি?

উত্তর:

```

class Calculation extends Exception {
    public Calculation() {
        System.out.println("Calculation class is instantiated");
    }
    public void add(int a, int b) {
        System.out.println("The sum is "+(a+b));
    }
}
  
```

public class Main {

 public static void main(String []args){

 }

 }

Output:

Calculation class is instantiated

The sum is 30

Explanation: try ব্রক থেকে ক্যালকুলেশনের object টি throw করা হয়, ক্যাচ ব্রকে ধরা পড়ে। এই ক্লাসের অবজেক্ট ব্যবহার করে

ক্যালকুলেশন ক্লাসের অ্যাড() 10 এবং 20 পূর্ণসংখ্যার মান দিয়ে called করা হয়। তাই যোগফল 30 সুন্দরভাবে হয়। মেইন ক্লাসের অবজেক্ট কেবল তখনই throw করা যেতে পারে যখন object এর ধরনটি throwable হয়। এটি করার জন্য, আমাদের throwable class extend করতে হবে।

প্রশ্ন ১৬. নিচের প্রয়োজনের আউটপুট কোনটি?

উত্তর:

```

public class Main {
    void a() {
        try{
            System.out.println("a(): Main called");
            b();
        }catch(Exception e) {
            System.out.println("Exception is caught");
        }
    }
    void b() throws Exception {
        try{
            System.out.println("b(): Main called");
            c();
        }catch(Exception e){
            throw new Exception();
        }
        finally {
            System.out.println("finally block is called");
        }
    }
    void c() throws Exception {
        throw new Exception();
    }
    public static void main (String args[]){
        Main m = new Main();
        m.a();
    }
}
  
```

Output:

a(): Main called

b(): Main called

finally block is called

Exception is caught

Explanation: main method এ, Main এর a() কে called করা হয় যা একটি বার্তা প্রিন্ট করে এবং b() কল করে। b() কিছু বার্তা প্রিন্ট করে এবং তারপর c() কল করে। c() একটি exception throw করে যা মেছেড b এর ক্যাচ ব্রক দ্বারা execute হয়। যাইহোক, এটি মেছেড a() দ্বারা execute করার জন্য throw Exception() ব্যবহার করে এই exception টি প্রচার করে। আমরা জানি, finally ব্রক সর্বদা execute হয় তাই মেছেড b() তে finally ব্রকটি প্রথমে কার্যকর করা হয়। এবং একটি বার্তা প্রিন্ট করে। শেষ পর্যন্ত, exception টি মেছেড a() এর ক্যাচ ব্রক দ্বারা পরিচালিত হয়।

प्रश्न ५७. कठखला अवजोड़ी टेकि यावे निवे कोत येते ?

```
1. String s1="Welcome";
2. String s2="Welcome";
3. String s3="Welcome";
```

Only one object will be created using the above code because strings in Java are immutable.

प्रश्न ५८. कठखला अवजोड़ी टेकि यावे निवे कोत येते ?

```
String s = new String("Welcome");
Two objects, one in string constant pool and other in non-pool(heap).
```

प्रश्न ५९. निवे योग्यावे आटोपूट देव करना?

```
public class Test {
    public static void main (String args[]) {
        String a = new String("Harun is a good player");
        String b = "Harun is a good player";
        if(a == b) {
            System.out.println("a == b");
        }
        if(a.equals(b)) {
            System.out.println("a equals b");
        }
    }
}
```

Output: a equals b

Explanation: अवजोड़ी "==" नुटि टिक्की अवजोड़ी रेखावेल समान किना ता परीक्षा करते। यासि उत्तर टिक्की एकी contain शाळ करते, तिक्का तासेरे रेखावेल समान नव करते उत्तरही तिक्का उपाय देवये (कम्प्यूटरी एवं String literal) तारी, a == b असाव, असावावे, equal() method सर्वांगा contain परीक्षा करते। यासि तासेरे contain समान तारी a equals b print हवते।

प्रश्न ६०. निवे योग्यावे आटोपूट देव करना?

```
public class Test{
    public static void main (String args[])
    {
        String s1 = "Harun is a good player";
        String s2 = new String("Harun is a good
player");
        s2 = s2.intern();
        System.out.println(s1 == s2);
    }
}
```

Output:

Explanation: intern() method ए टिक्का पूल देवके टिक्की अवजोड़ी रेखावेल प्राप्त करते। एक टेक्के, string literal यावहार करते s1 टेक्की किंवा यावे येतावा, टिक्का पूल यावहार करते s2 टेक्की किंवा यावे s2 एवं तेक्कावेल परिवर्तित हवते, एवं अप्पारेटर == true निटीन करते।

प्रश्न ६१. एकी वाक्ये कठखला शब्द आवे देवी गदावे करावे जन्य एकी योग्यावे निवून ?

उत्तर:

```
public class Test {
    public static void main (String args[])
    {
        String s = "Sharma is a good player and he is so
punctual";
        String words[] = s.split(" ");
        System.out.println("The Number of words present in
the string are :" +words.length);
    }
}
```

Output: The Number of words present in the string are : 10

प्रश्न ६२. निवे योग्यावे आटोपूट देव करना?

```
import java.util.regex.*;
class RegexExample2{
    public static void main(String args[]){
        System.out.println(Pattern.matches("s", "as")); //line 4
        System.out.println(Pattern.matches("s", "mk")); //line 5
        System.out.println(Pattern.matches("s", "mst")); //line 6
        System.out.println(Pattern.matches("s", "amms")); //line 7
        System.out.println(Pattern.matches("s", "mas")); //line 8
    }
}
```

Output:

true

false

false

false

true

Explanation: टिक्की-एवं विटीर अवजवे s यावहार नाहीन ता true निटी करते, ताईन ता false निटी करते येतेवै विटीर अवजवती ता नव, ताईन ता false निटी करते काळ टिक्की यावे एवं टिक्की अवजव नाहीन, ताईन ता false निटी करते येतेवै टिक्की-एवं अवजव एवं तेक्की-एवं ताईन ता true निटी करते येतेवै तिक्की अवजवती s :

प्रश्न ६३. Finalize() method नव काळ टिक्की?

उत्तर:

finalize() method ए object garbage collected रावावे उक्के अवजव करते हवते। एकी तिक्कावे अवजव सम्पादन करते अवजवत हवते। Consider the following example.

public class FinalizeTest {

```
int j=12;
void add() {
    j=j+12;
    System.out.println("J=" +j);
}
public void finalize() {
    System.out.println("Object is garbage collected");
}
public static void main(String[] args) {
    new FinalizeTest().add();
    System.gc();
    new FinalizeTest().add();
}
}
```

प्रश्न ६४. आजाते काळ उपाये इनपूट नेवा याव?

उत्तर: In Java, there are three ways by using which, we can take input from the console.

o Using BufferedReader class: Consider the following example.

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class Person {
    public static void main(String[] args) throws IOException {
        System.out.println("Enter the name of the person");
        BufferedReader reader = new BufferedReader(new
InputStreamReader(System.in));
        String name = reader.readLine();
        System.out.println(name);
    }
}
```

o Using Scanner class: Consider the following example.

```
import java.util.*;
public class ScannerClassExample2 {
    public static void main(String args[])
    {
        String str = "Hello/This is JavaPoint/My
name is Abhishek.";
        //Create scanner with the specified String
        Object
        Scanner scanner = new Scanner(str);
        System.out.println("Boolean Result:
"+scanner.hasNextBoolean());
        //Change the delimiter of this
scanner
        scanner.useDelimiter("/");
        //Printing the tokenized Strings
        System.out.println("---Tokenizes String---");
    }
}
```

```
while(scanner.hasNext()){
    System.out.println(scanner.next());
}
//Display the new delimiter
System.out.println("Delimiter used:
"+scanner.delimiter());
scanner.close();
}
```

o Using Console class: Consider the following example.

```
import java.io.Console;
class ReadStringTest{
    public static void main(String args[]){
        Console c=System.console();
        System.out.println("Enter your name: ");
        String n=c.readLine();
        System.out.println("Welcome "+n);
    }
}
```

प्रश्न ६५. Socket टिक्की?

उत्तर: Socket अप्पिलेर यावे योग्यावेले अवजव एकी endpoint ; एकी TCP यावहार करते नुटि कम्प्यूटरावेल संयोग करावा यावहार प्रवाह करते। संकेत तासेरे एकी संकेत टेक्की करावे यावहार करते हवते।

प्रश्न ६६. Wrapper classes टिक्की?

उत्तर: wrapper झालावले एवं class ए विचित्र data ताईनले अवजवती हिसाबे आजाते करावे अवजवत हवते।

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

प्रश्न ६७. Autoboxing and unboxing टिक्की?

उत्तर: autoboxing एवं विचित्र एकी टाइप तालोके corresponding wrapper class object ए कम्प्यूटरी करावा याविना। unboxing एवं wrapper class object तालोके corresponding primitive data type ए कम्प्यूटरी करावीना।

प्रश्न ६८. निवे योग्यावे आटोपूट देव करना?

```
public class Test1 {
    public static void main(String[] args) {
        Integer i = new Integer(201);
        Integer j = new Integer(201);
        if(i == j) {
    }
}
```

```

        System.out.println("hello");
    }
    else {
        System.out.println("bye");
    }
}

```

Output: bye

Explanation: Integer class -127 থেকে 127 পর্যন্ত পূর্ণসংখ্যার মানগুলি cache করে। তাই, পূর্ণসংখ্যা object গুলি তথ্যাত -128 থেকে 127 মধ্যে তৈরি করা যেতে পারে। অপারেটর == 127-এর বেশি মানের জন্য কাজ করবে না; তাই bye প্রিন্ট করে।

প্রশ্ন ৬৯. Singleton class কি?

উত্তর: সিঙ্গেলটন ক্লাসটি এমন এক ক্লাস যা একাধিকবার instantiated করা যায় না। ক্লাস সিঙ্গেলটন বানাতে, আমরা হয় এর constructor কে প্রাইভেট করব বা স্ট্যাটিক getinstance মেথড ব্যবহার করব। Consider the following example.

```

class Singleton{
    private static Singleton single_instance = null;
    int i;
    private Singleton () {
        i=90;
    }
    public static Singleton getInstance() {
        if(single_instance == null) {
            single_instance = new Singleton();
        }
        return single_instance;
    }
}

public class Main {
    public static void main (String args[]) {
        Singleton first = Singleton.getInstance();
        System.out.println("First instance integer value:"+first.i);
        first.i=first.i+90;
        Singleton second = Singleton.getInstance();
        System.out.println("Second instance integer value:"+second.i);
    }
}

```

প্রশ্ন ৭০. How to perform Bubble Sort in Java?

উত্তর: Consider the following program to perform Bubble sort in Java.

```

public class BubbleSort {
    public static void main(String[] args) {
        int[] a = {10, 9, 7, 101, 23, 44, 12,
                  78, 34, 23};
        for(int i=0;i<10;i++) {
            for (int j=0;j<10;j++) {
                if(a[i]<a[j]) {

```

```

                    int temp = a[i];
                    a[i]=a[j];
                    a[j] = temp;
                }
            }
            System.out.println("Printing Sorted List ...");
            for(int i=0;i<10;i++) {
                System.out.print(a[i]);
            }
        }
    }

```

Output:

Printing Sorted List ...
7 9 10 12 23 34 34 44 78 101

প্রশ্ন ৭১. How to perform Binary Search in Java?

উত্তর: Consider the following program to perform the binary search in Java.

```

import java.util.*;
public class BinarySearch {
    public static void main(String[] args) {
        int[] arr = {16, 19, 20, 23, 45, 56, 78, 90, 96, 100};
        int item, location = -1;
        System.out.println("Enter the item which you want to search");
        Scanner sc = new Scanner(System.in);
        item = sc.nextInt();
        location = binarySearch(arr,0,9,item);
        if(location != -1)
            System.out.println("the location of the item is "+location);
        else
            System.out.println("Item not found");
    }

    public static int binarySearch(int[] a, int beg, int end, int item) {
        int mid;
        if(end >= beg) {
            mid = (beg + end)/2;
            if(a[mid] == item) {
                return mid+1;
            }
            else if(a[mid] < item) {
                return binarySearch(a,mid+1,end,item);
            }
            else {
                return binarySearch(a,beg,mid-1,item);
            }
        }
    }
}

```

```

        }
        return -1;
    }
}

```

Output: Enter the item which you want to search 45
the location of the item is 5

প্রশ্ন ৭২. Thread এর শাইক সাইকেল এর states কোন লিখ?

1. New
2. Runnable
3. Running
4. Waiting/Blocked
5. Dead/Terminated

প্রশ্ন ৭৩. জাভাতে কলেকশন কি?

উত্তর: Collection is a framework that is designed to store the objects and manipulate the design to store the objects.

Collections are used to perform the following operations:

- Searching
- Sorting
- Manipulation
- Insertion
- Deletion

A group of objects is known as collections. All the classes and interfaces for collecting are available in Java util package.

প্রশ্ন ৭৪. Let's N is a Number, write a java program to check N is Armstrong or not? [BREB Assistant Junior Engineer (IT) Written-2019]

উত্তর:

```

import java.util.Scanner;
Public class Armstrong {
    Public static void main (String [] args) {
        Int n, number, rem, temp=0;
        System.out.println("enter n=");
        Scanner scanner = new Scanner (System.in);
        n = scanner.nextInt ();
        number1=n;
        number2=n;
        while(n){
            n=n/10;
            coun++;
        }
        While(number1){
            rem = number1% 10;
            number1= number1/ 10;
        }
    }
}

```

```

        temp=temp+pow(rem,count);
    }
    If( temp == number2 )
        System.out.println("Armstrong");
    else System.out.println("Not Armstrong");
}
}

```

নিচের প্রোগ্রামের আউটপুট কোনটা করবে?

Program 1:

```

class Q1{
    public static void main(String args[]){
        int a=4;
        int b=8;
        if(b-a--)>=a) {
            if((b+a)%2==1)
                System.out.println(a*b);
            else
                System.out.println(b+a);
        }
    }
}

```

Output: 24**Program 2:**

```

class Q2{
    public static void main(String args[]){
        int []A=new int[8];
        int i=0;
        for(i=1;i<A.length-1;){
            A[++i]=i;
        }
        String res="" +A[2]+(4%2)+(5%2)+i;
        System.out.print(res);
    }
}

```

Output: 2017**Program 3:**

```

class A{
    int x=10;
}
class B extends A{
    int x=20;
}
public class Q3{
    public static void main(String []args){
        A a=new B();
        System.out.print(a.x);
    }
}

```

Output: 10

Reason: Derived class constructor first invokes the base class constructor.

Program 4:

```
class Q4{
    public static void main(String args[]){
        int i=3;
        int e=6;
        int x=7;
        String s="";
        if((i+e)>=(i+x)){
            i++;
            s="gg";
            e--;
        }
        else if((i+e+x)>=15){
            x++;
            s="wp";
        }
        System.out.print(x+s);
    }
}
```

Output: 8wp

4(a). What is the output of the following Java program? [Islami Bank Officer(Software Engineer) 2019]

```
public class Test
{ Test (int a, int b){
System.out.println ("a = " + a + b = " + b);
}
Test (int a, float b)
{ System.out.println ("a=" + a + "b" + b);
}
public static void main(String[] args){
byte a = 10;
byte b = 15;
Test test = new Test (a, b);
}
}
```

Output: a=10 b=15

4(b) What is the output of the following Java program?

```
class Test{
    int i;
}
public class Main{
    public static void main(String args[]){
Test test = new Test();
}
}
```

```
System.out.println (test.i);
}
}
```

Output: 0

```
(c) class Test{
int test_a, test_b;
Test (int a, int b){
test_a = a;
test_b = b;
}
public static void main(String args[])
{ Test test = new Test ();
System.out.println (test.test_a + b
"+test.test_b);
}
}
```

Output: Error, require int, int argument in constructor but not found

Program 5:

```
class Q5{
    public static void main(String args[]){
String s1=new String("11001");
int d=Integer.parseInt(s1,2);
System.out.print(d);
}
}
```

Output: 25**Program 6:**

```
class Base{
Base() {
System.out.println("Base");
}
Base(int n) {
System.out.println("Base:" +n);
}
class Derive extends Base{
Derive() {
super(10);
System.out.println("Derive");
}
Derive(int n,int m) {
super(n);
System.out.println("Derive:" +n +"," +m);
}
}
```

class Q6{

```
public static void main(String args[]){
Derive D1=new Derive();
Derive D2=new Derive(40,50);
}
}
```

Output:

Base:10
Derive
Base:40
Derive:40,50

7. Find the output of following java program.

```
class MainClass{
    public static void main(String arg[])
{ int arr[][]={{4,3},{2,1}};
int i,j;
for(i=1;i>-1;i--){
    for(j=1;j>-1;j--){
        System.out.print(arr[i][j]);
    }
}
}
```

Output: 1234**8. Find the output of following java program.**

```
class access{
static int x;
void increament(){
    ++x ;
}
}
class MainClass{
    public static void main (String arg[])
{ access obj1 = new access();
access obj2 = new access();
obj1.x = 0;
obj1.increament();
obj2.increament();
System.out.print(obj1.x +
obj2.x);
System.out.println();
}
}
```

Output: 4**9. Find the output of following java program.**

```
class MainClass{
    public static void main (String arg[])
{ System.out.print('h' + 'i');
System.out.println();
}
}
```

Output: 209**10. Find the output of following java program.**

```
class MainClass{
    public static void main (String arg[]) {
int i;
for( i=1; i++ ){
    System.out.print(i);
    break;
}
System.out.println();
}
}
```

Output: compilation error.

Explanation: C এবং C++ এর মত, আমরা জাভার ক্ষেত্রে কভিনেন চেক করার জন্য ০ এবং ১ ব্যবহার করতে পারি না। জাভাতে আমাদের শর্টসাপেক অপারেটরগুলির অবশ্য পরীক্ষা করার জন্য বুল ডেরিয়েক্স 'true' এবং 'false' ব্যবহার করতে হবে।

The correct code should be:

```
class MainClass{
    public static void main (String arg[])
{ int i;
for( i=1; true; i++ ) {
    System.out.print(i);
    break;
}
System.out.println();
}
}
```

11. Find the output of following java program.

```
class MainClass{
    public static void main (String arg[])
{ System.out.print( func() );
System.out.println();
}
int func() {
    int test = 100;
    return test;
}
}
```

Output: compilation error**The correct code should be:**

```
class MainClass{
    public static void main (String
    arg[])
    {
        System.out.print( func() );
        System.out.println();
    }

    static int func()
    {
        int test = 100;
        return test;
    }
}
```

(b)

```
class Test{
    public static void main(String args)
    { for(int i=0; 0; i++)
        System.out.println("Hello Javatpoint");
    }
}
```

Output: Error**প্রশ্ন 12. Static public void এর পরিবর্তে public static void লিখলে কি হবে?****উত্তর:** প্রয়োজনি সম্পর্কের জন্য আভাস করে সঠিকভাবে run হবে কারণ স্পেসিফিকেশনের ক্ষেত্রে কিছু যাই আসে না।**প্রশ্ন 13. Static methods and variables কি? [Islami Bank Officer(software engineer)-2019]****উত্তর:** Java তে Variable আর Method ডিজিটেল Object এর অধীনে থাকে। Object instantiate না করে কেন variable করা যায় না বা method call করা যায় না (যদি কোন 'variable' বা 'method' কে static হিসেবে declare করা হয় তাহলে সেটা object এর অধীনে না থেকে class এর অধীনে থাকে)। সেই class এর যত Object instantiate করা হোক না কেন, static variable একটাই থাকে, এমনকি যেকোন static member (Variable/method) class এর object instantiate না করেও করা যায়। (সরাসরি class reference থেকে)যদি কোন class এ static block থাকে, Java Virtual Machine class loading/linking/initializing এর সময়ে সেই static block এর statement execute করে।

```
class StaticTest {
    private static int n = 10;
    private static int m = 20;
    static {
        System.out.println("Value of n : " + n);
    }

    public static int getM() {
        return m;
    }
}
```

```
}
public static void main(String []args) {
    System.out.println("Value of m : " +
    StaticClass.getM());
}
```

প্রশ্ন 18. Instance কি?**উত্তর:** কোনো ক্লাস এর যখন অবজেক্ট initialization করা হয় তিক তখন ওই ক্লাস এর একটা instance virtual machine এ save হয়। মানে ওই ক্লাস টার একটা কপি ভার্চুয়াল মেশিন এ save হয়। এটাকে instance বলে।**প্রশ্ন 15. Static (class) method and instance method এর মধ্যে পার্থক্য কীভুলি লিখ?****উত্তর:**

Static Or Class Method	Instance Method
স্ট্যাটিক হিসাবে ঘোষিত একটি মেথড স্ট্যাটিক মেথড হিসাবে পরিচিত।	স্ট্যাটিক হিসাবে ঘোষনা করা হয়নি এমন মেথডকে instance method বলা হয়।
static methods কল করার জন্য অবজেক্ট তৈরি করার জন্য অবজেক্ট তৈরি করার দরকার নেই।	instance method কল করার জন্য অবজেক্ট তৈরি করার জন্য অবজেক্ট তৈরি করার দরকার আছে।
স্ট্যাটিক কলটারে (static method, static block, and static nested class) নন-স্ট্যাটিক মেথডকে সরাসরি একক্সেস করা যায়।	স্ট্যাটিক, নন-স্ট্যাটিক মেথডকে সরাসরি একক্সেস করা যায়।
উদাহরণ বরুপ: public static int cube(int n){ return n*n*n;}	উদাহরণ বরুপ: public void msg(){...}.

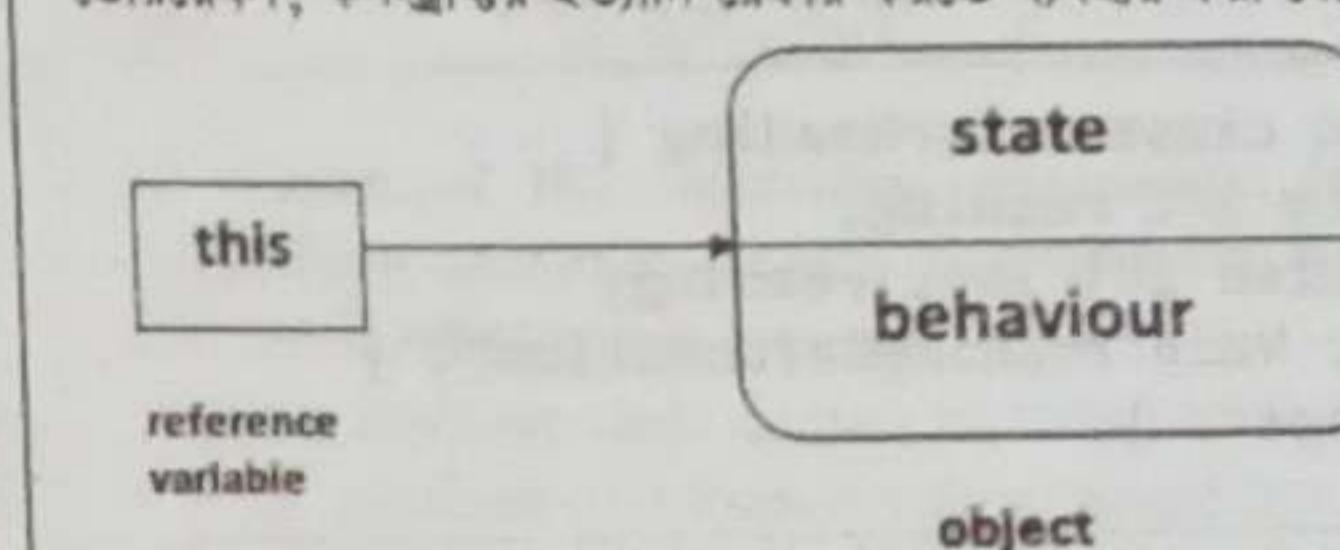
প্রশ্ন 16. Constructor কি static করা যায়?**উত্তর:** যেহেতু আমরা জনি যে static context (মেথড, ব্রক, বা ডেরিভেক্ট) ক্লাসের অঙ্গত, অবজেক্টের নয়। যেহেতু কনস্ট্রাক্টরগুলি কেবল তখনই কল হয় যখন অবজেক্ট তৈরি করা হয়, তাই কনস্ট্রাক্টরকে স্ট্যাটিক করার কোনও দরকার নেই। তবে, আপনি যদি এটি চেষ্টা করেন তবে কম্পাইলার একটি কম্পাইলার এর দেখাবে।**প্রশ্ন 17. Can we make the abstract methods কি static করা যায়?****উত্তর:** জাভাতে, আমরা যদি abstract method কে যদি static করি তবে এটি ক্লাসের অংশ হয়ে যাবে এবং আমরা একে সরাসরি কল করতে পারি যা অ্যাপ্লিকেশন। আনডিফাইল মেথড কল করা সম্পূর্ণ useless তাই এটি allowed নয়।**প্রশ্ন 18. Abstraction কি?****উত্তর:** abstraction হলো ইম্প্রিমেটিশন ডিটেইলস হাইড করে ব্যবহারকারীর কাছে কেবল ফাংশনালিটি দেখানোর প্রক্রিয়া। এটি ব্যবহারকারীর কাছে কেবল প্রয়োজনীয় জিনিসগুলি প্রদর্শন করে এবং অভ্যন্তরীণ তথ্য গোপন করে। উদাহরণবরুপ, আপনি এসএমএস প্রেরনের সময় তথ্যাত্মক এসএমএস টাইপ করেন এবং বার্তা প্রেরণ করেন, আপনি**এসএমএস প্রেরন সম্পর্কে অভ্যন্তরীণ প্রক্রিয়াজাতকরণ জানেন না। abstraction আপনাকে কীভাবে এটি করে তার পরিবর্তে অবজেক্টটি কী করে তাতে মনোনিবেশ করতে সক্ষম করে।****In Java, there are two ways to achieve the abstraction.**

- Abstract Class
- Interface

প্রশ্ন 19. Abstraction and encapsulation এর মধ্যে পার্থক্য কীভুলি লিখ?**উত্তর:** abstraction implementation details কে hide করে যেখানে encapsulation কোড কে wraps করে এবং ডাটাকে একটি সিলেক্স ইউনিটে পরিণত করে।**প্রশ্ন 20. Abstract class কী static variables and methods declare করা যায়?****উত্তর:** হ্যাঁ, আমরা abstract method এ স্ট্যাটিক ভেরিয়েবল এবং স্ট্যাটিক মেথড ঘোষণা করতে পারি; যেহেতু আমরা জনি যে static context অ্যাক্সেস করার জন্য কোনও অবজেক্ট তৈরি করার কোনও প্রয়োজন নেই, তাই আমরা abstract class name ব্যবহার করে abstract class এর অভ্যন্তরে ঘোষিত static context কে অ্যাক্সেস করতে পারি।

```
abstract class Test{
    static int i = 102;
    static void TestMethod() {
        System.out.println("hi !! I am good !!");
    }
}
```

```
public class TestClass extends Test{
    public static void main (String args[])
    {
        Test.TestMethod();
        System.out.println("i = "+Test.i);
    }
}
```

প্রশ্ন 21. Abstract class কি instantiated করা যায়?**উত্তর:** না, abstract class কখনই instantiated করা যায় না যদিও তাতে কোন কনস্ট্রাক্টর থাকে এবং এর সমস্ত মেথডের ইম্প্রিমেটিশন করা হয়।**প্রশ্ন 22. Abstract class ছাড়া কি abstract method ডিক্রিয়ার করা যায়?****উত্তর:** না, কোনও ক্লাসে যদি abstract method থাকে তবে সেই ক্লাসটি অবশ্যই abstract হবে।**প্রশ্ন 23. This keyword কেন ব্যবহার করা হয়?****উত্তর:** this keyword মূলত নিজ ক্লাসের অবজেক্টকে রেফারেন্স করে। this কীওয়ার্ড একটি রেফারেন্স ভেরিয়েবল যা বর্তমান অবজেক্টকে বোঝায়। জাভাতে এই কীওয়ার্ডের বিভিন্ন ব্যবহার রয়েছে। যেমন this keyword বর্তমান ক্লাসের বৈশিষ্ট্য যেমন মেথড,**ভেরিয়েবল, কনস্ট্রাক্টর ইত্যাদি রেফার করতে ব্যবহার করা যেতে পারে।****প্রশ্ন 24. This keyword এর ধারণা ব্যবহার কি কি? [Islami Bank Officer(software engineer)-2019]****উত্তর:**

- this keyword কারেট ক্লাসের instance variable কে রেফার করতে ব্যবহার করা যেতে পারে।
- this keyword কারেট ক্লাসের মেথডকে কল করতে ব্যবহার করা যেতে পারে।
- this keyword কারেট ক্লাসের constructor কে কল করতে ব্যবহার করা যেতে পারে।
- this keyword কে মেথডের আর্গুমেন্ট হিসাবে pass করা যেতে পারে।
- this keyword কে constructor এর আর্গুমেন্ট হিসাবে pass করা যেতে পারে।
- this keyword দিয়ে কারেট ক্লাসের instance কে মেথড থেকে রিটার্ন করা যেতে পারে।

প্রশ্ন 25. Web Development এ Java Script এর প্রয়োজিয়তা সম্পর্কে বিবরণ দিন। [ICT division (AP)-2021]**উত্তর:** JavaScript যেটাকে আমরা সংক্ষেপে JS নামেই চিনি, যা একটি লাইট গ্রেট, ইন্টারপ্লেটেড (অর্থাৎ কম্পাইল করার প্রয়োজন নেই) এবং Prototype-based, মাল্টি প্যারাডাইম, ডাইনামিক ক্রিপ্টিং ল্যাঙ্গুেজ যা একাধারে Object-oriented, Imperative, এবং Declarative তথা Functional Programming স্টাইল সাপোর্ট করে।**যদিও জাভাক্রিপ্টকে একটি লাইট সাইড ক্রিপ্টিং ল্যাঙ্গুেজ বা প্রাইজার ক্রিপ্টিং ল্যাঙ্গুেজে বলা হয় কিন্তু বর্তমানে জাভাক্রিপ্ট ক্লায়েন্ট সাইড ও নেটওয়ার্ক প্রয়ার্থি, মোবাইল ও ডেক্টপ্যাক্সেমেন্ট, পেইমেন্ট ডেভেলপমেন্ট থেকে তরু করে ইন্টারনেটে অব থিস, মেশিন লার্নিং সহ আরো অনেক কাজেই ব্যবহৃত হচ্ছে।****প্রশ্ন 26: In Java program. Write the method in given box for the Electric bill calculation if unit is less than 100 then unit rate 4.0 taka and after 100 unit rate is 5.50 and reaming unit rate is 6.00.****উত্তর:** একবা বক্স এ কোডের অংশ লিখে ইলেক্ট্রিক বিল ক্যালকুলেশনের জন্য একটি জাভা প্রয়োজ লিখ। যেখানে প্রথম 100 ইউনিট 4 টাকা হারে, পরের 100 ইউনিট 5.50 টাকা হারে এবং বাকি reaming unit 6 টাকা হারে হিসেব করতে হবে। [BPDB-AE-2021]**[Bill rate 4.0 if unit<= 100****Bill rate 5.50 if (unit >100 && unit <=200)**

Bill rate 6.00 for remaining units.]

```
public class meterReading {
    private int reading;
    protected int last_reading;
    public void readingCalculation() {
        reading++;
    }

    public double getUnit(){
        return last_reading-reading;
    }
}
```

```
public class BillerClass extends meterReading {
    double bill;
    public double getBill(){
        last_reading=getUnit();
        if(last_reading<100)
            bill=last_reading*4.00;
        else if(last_reading<200)
            bill=100*4.00+(last_reading-100)*5.50;
        else if(last_reading>=200)
            bill=100*4.00+100*5.50+(last_reading-200)*6.00;
        return bill;
    }
}
```

প্র ২৭. What are the differences between C++ and Java programming language? [C++ এবং Java এর মাঝে পার্থক্য লিখ] [০৬তম বিসিএস]

উত্তর: নিচে C++ এবং Java এর মাঝে পার্থক্য দেওয়া হলো:

C++	Java
C++ কম্পাইলার ব্যবহার করে থাকে।	Java কম্পাইলার এবং ইন্টেরপ্রেটর দ্বাইটাই ব্যবহার করে থাকে।
অপারেটর ভারপোড়িং এবং মেথড ভারপোড়িং উভয়ই সাপোর্ট করে।	তথ্যাজ মেথড ভারপোড়িং সাপোর্ট করে।
New এবং Delete Keyword দ্বারা Manual Object Management করে থাকে।	Java তে আছে Built-In Automatic Garbage Collection.
স্ট্রাকচার, ইউনিভ, মাল্টিপ্ল ইনহেরিটেন্স সাপোর্ট করে।	স্ট্রাকচার, ইউনিভ, মাল্টিপ্ল ইনহেরিটেন্স সাপোর্ট করে না।
Platform dependent	Platform independent

প্র ২৮. Write program In Java/C to find out the factorial of a number using recursion. Also write its iterative equivalent program. [একটি নামাবের

ফ্যাক্টরিয়েল ভালু বের করার জন্য রিকর্সিভ প্রোগ্ৰাম সি/জাভাতে লিখ। [খাদ্য মন্ত্রণালয়-এপি-২০২১]

উত্তর: Recursive Program: [যেহেতু সবাই C programming মৌলিক পারি, তাই Java programming দেয়া হল এতে করে Java একটু বিভিন্ন হল।]

```
import java.util.*;
import java.lang.*;
import java.io.*;

public class Factorial {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int num = sc.nextInt();
        long factorial = multiplyNumbers(num);
        System.out.println("Factorial of " + num + " = "
                + factorial);
    }

    public static long multiplyNumbers(int num) {
        if (num >= 1)
            return num * multiplyNumbers(num - 1);
        else
            return 1;
    }
}
```

Iterative Program:

```
import java.util.*;
import java.lang.*;
import java.io.*;
public class Factorial {
    public static void main (String []
    args) {
        Scanner sc= new Scanner (System.in);
        int num = sc. nextInt ();
        long factorial = 1;
        for (int i = 1; i <= num; ++i) {
            factorial *= i;
        }
        System.out.printf("Factorial of
%d = %d", num, factorial);
    }
}
```

প্র ২৯. Complete the following Java Program. [নিচের জাভা প্রোগ্রামটি কম্পিউট করা] [BCC-AP-2021]

```
class A {
    int alpha;
    int beta;
    public A(int alpha, int beta) {
        this.alpha = alpha;
        this.beta = beta;
    }

    public void display(){
        System.out.println("Alpha " + alpha +
"\nBeta " + beta);
    }
}
```

```
class Gamma extends A{
    @Override
    public void display()
    {
    }
}

public class Main {
    public static void main(String[] args)
    {
        Gamma g = new Gamma (3, 30, 10);
        g.display();
    }
}
```

Ans:

```
i:
int gamma;
public Gamma (int alpha, int beta,
int gamma){
    super(alpha, beta);
    this.gamma = gamma;
}

ii:
super.display();
System.out.println("Gamma " +
gamma);
```

চাকুরিতে আসা কিছু এমনিকিউ

৩. Which of the following is not a method of the Thread class? [কোনটি প্রেত ক্লাসের মেথড নয়?] [Combine bank-AP-2021]

- a) Sleep (long msec)
- b) stop ()
- c) go ()
- d) yield ()

Explanation:

Thread methods		
setName()	getThreadGroup()	checkAccess()
getName()	interrupt()	currentThread()
getPriority()	interrupted()	dumpStack()
isAlive()	isAlive()	getId()
join()	isDaemon()	getState()
Stop()	sleep()	setDaemon(boolean on)
start()	setPriority(int newPriority)	
activeCount()	yield()	

২. Which of the following statements is correct regarding abstract classes? [আবস্থার দ্রুত সম্পর্কিত কোন স্টেটমেন্ট টি সঠিক?] [Combine bank-AP-2021]

- a) An abstract class cannot be extended
- b) A subclass of a non-abstract superclass cannot be abstract
- c) A subclass can override a concrete method in a superclass to declare it abstract
- d) An abstract class cannot be used as a data type

Explanation: An abstract class may:

- An abstract class can be extended.
- A subclass of a non-abstract superclass can be abstract.
- A subclass can override a concrete method in a superclass to declare it abstract.
- An abstract class can be used as a data type.

৩. What is the output of this Java program? [নিচে জাভা প্রোগ্রামের আউটপুট নির্ণয় করুন] [Combine bank-AP-2021]
[Sonali Bank Ltd.Assistant Database Administrator-2020]

```
class Test {
    int i;
    public class Main {
        public static void main(String args[]){
            Test t= new Test();
            System.out.println(t.i);    }
    }
}
```

- a) 0
- b) A garbage value
- c) The program will cause a runtime exception because the variable 'i' was not initialized
- d) The program will cause an compile error because the object 't' was not initialized

Explanation: জাভাতে, ক্লাস এবং অবজেক্টের যে ক্ষেত্রগতিতে কোনও স্পষ্ট ইনিশিয়ালাইজার নেই এবং আরে উপাদানগুলি ঘোঁঞ্জিয়া ভাবে তাদের প্রকারের জন্য ডিফল্ট মান দিয়ে আরম্ভ করা হয় (বুলিয়ানের জন্য false , all numerical types জন্য 0 , all reference types জন্য null)। জাভাতে Local variable গুলি অবশ্যই আরেম্ব করার আগে অবশ্যই assigne করা উচিত, বা এটি একটি কম্পাইল তুটি।

৪. What is the output of this java program? [নিচের জাভা প্রোগ্রামের আউটপুট নির্ণয় করুন] [Combine 4 bank -AP-2020]

```
class Test{
    int i;
}
```

```
public class Main{  
    public static void main(String args[]){  
        Test t;  
        System.out.println(t.i);  
    }  
}
```

- (A) 0
 - (B) A garbage Value
 - (C) The program will cause an runtime exception because the variable i was not initialized
 - (D) The program will cause an compile error because the object t was not initialized.

Q. Interfaces in Java are meant to be _____. [১০টি
ইন্টারফেস কল্পে মোকাবে হবে _____.] [Combine 4 bank - AP -
2020]

- (A) Extended
 - (B) Implemented
 - (C) Overridden
 - (D) Used by creating object

৬. Which of the following statements is/are true about inheritance in java [জাভাতে ইন্হেরিটেন্স সম্বন্ধে কোন কোট্টেক্ষণি সত্য] [Combine 4 bank -AP -2020]

1. Private methods are final
2. Protected methods are final
3. Private methods cannot be overridden
4. Protected members of a class are accessible by inherited classes of another package.

(A) 1,3,4 (B) 1,3 (C) 2,3,4 (D) 2,4

(c) Kotlin (d) Android is not an operating system

৭. Which of the following statements is not true for java language[জাভাটে কেন স্টেটমেন্টি সত্য নয়]? [Combine 4 bank -AP -2020]

- (A) The number 1 can be used instead of the keyword True
 - (B) Trying to store a fraction value in an int datatype causes compiler error
 - (C) Static members of a class can be accessed without creating objects of that class
 - (D) If not specified otherwise, the initial value of an integer variable is 0

v. Java uses a keyword _____ to preface a block of code that is likely to cause an error condition and throw an exception. [একটি ত্রুটির কোড প্রিফেস করার জন্য আমা একটি কীওয়ার্ড _____ ব্যবহার করে যা সম্ভবত cause এবং error condition এবং একটি exception throw করে।] [BSC
2 Combined Bank - Senior Officer (IT)-2020]

- a) throw b) catch
c) finally d) try

Which is not the feature of JAVA OOP? [জেন
স্টার্ট কোড নথি] [Janata Bank-ANE-2020]

- o. Which information is not correct for any constructor of a java class? [যে কোন কলস্ট্রাকটরের জন্য কোন তাত্ত্বিক জ্ঞান সংরক্ষণ করা হবে না এবং কোন ইনকোর্পোরেশন করা নথি] [Sonali Bank Ltd.]

Assistant Database Administrator-2020]

- a) Constructor is not inherited
 - b) Constructor has no return type
 - c) Constructor can be final
 - d) Constructor can be overloaded

১১. Which language was used to build android operating system? [আন্ড্রয়েড অপারেটিং সিস্টেম টেক্নিকে কোন প্রাথমিক ব্যবহার হয়?] [Sonali Bank Ltd. Assistant Database Administrator-2020]

- (a) Java (b) Python
(c) Kotlin (d) Android is not an operating system

四

۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰	۱۱	۱۲
c	c	a	D	B	A	B	d	a	c	a	

Python Programming

Syllabus: Object Oriented Programming Basics; Class & Object, Properties Of Object Oriented Programming.]