ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग कॉन्सेप्ट्स
(शैक्षणिक वर्ष 2023-2024 से प्रभावी)
तृतीय सेमेस्टर
पाठ्यक्रम कोड CS322I3C CIA अंक 50
प्रति सप्ताह संपर्क घंटे की संख्या (L: T: P: S) 3:0:2:0 SEE अंक 50
शिक्षाशास्त्र के कुल घंटे 40L + 20P परीक्षा घंटे 03
क्रेडिट – 4
पाठ्यक्रम पूर्वापेक्षाएँ:

 जावा प्रोग्रामिंग का बुनियादी ज्ञान।
पाठ्यक्रम उद्देश्य:

 जावा की बुनियादी ऑब्जेक्ट-ओरिएंटेड विशेषताओं, कक्षाओं, ऑब्जेक्ट्स और इसके तरीकों को सीखें।
 सरल जावा प्रोग्राम बनाने, डीबग करने और चलाने के लिए जावा JDK वातावरण स्थापित करें।
 वंशानुक्रम, पैकेज और इंटरफेस की अवधारणाओं का पता लगाएँ।
 बहु-थ्रेडेड प्रोग्राम बनाएँ, इवेंट हैंडलिंग तंत्र
शिक्षण-अधिगम रणनीति:

पाठ्यक्रम वितरण के लिए शामिल की जा सकने वाली कुछ नमूना रणनीतियाँ इस प्रकार हैं:
 चाक और टॉक विधि/मिश्रित विधि
 पावर पॉइंट प्रस्तुति
 विशेषज्ञ वार्ता/वेबिनार/सेमिनार
 वीडियो स्ट्रीमिंग/स्व-अध्ययन/सिमुलेशन
 सहकर्मी से सहकर्मी गतिविधियाँ
 गतिविधि/समस्या आधारित शिक्षण
 केस अध्ययन
 MOOC/NPTEL पाठ्यक्रम
 पाठ्यक्रम सामग्री के संबंध में कोई अन्य नवीन पहल
पाठ्यक्रम सामग्री
मॉड्यूल - I
ऑब्जेक्ट ओरिएंटेड अवधारणाओं का परिचय: संरचनाओं की समीक्षा, प्रक्रिया-उन्मुख प्रोग्रामिंग प्रणाली, ऑब्जेक्ट ओरिएं
कक्षा और ऑब्जेक्ट: परिचय, सदस्य फ़ंक्शन और डेटा, ऑब्जेक्ट और फ़ंक्शन, ऑब्जेक्ट और ऐरे, नामस्थान, नेस्टेड कक्ष
पाठ्यपुस्तक 1: अध्याय 1: 1.1 से 1.9 अध्याय 2: 2.1 से 2.6 अध्याय 4: 4.1 से 4.2 8 घंटे
मॉड्यूल - II
जावा का परिचय: जावा का जादू: बाइट कोड, जावा डेवलपमेंट किट (JDK), जावा बजवर्ड्स, ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग, सर
पाठ्यपुस्तक 2: अध्याय: 1, 2, 3, 4, 5 8 घंटे
मॉड्यूल - III पृष्ठ 2 |
कक्षाएँ, वंशानुक्रम, अपवाद, पैकेज और इंटरफेस: कक्षाएँ: कक्षाएँ बुनियादी बातें; ऑब्जेक्ट घोषित करना; कंस्ट्रक्टर, यह क
वंशानुक्रम: वंशानुक्रम की मूल बातें, सुपर का उपयोग करना, बहु-स्तरीय पदानुक्रम बनाना, विधि ओवरराइडिंग।
अपवाद हैंडलिंग: जावा में अपवाद हैंडलिंग। पैकेज, एक्सेस सुरक्षा, पैकेज आयात करना, 8 घंटे
इंटरफेस।
पाठ्यपुस्तक 2: अध्याय 6,8, 9, 10
मॉड्यूल - IV
बहु-थ्रेडेड प्रोग्रामिंग: बहु-थ्रेडेड प्रोग्रामिंग: थ्रेड क्या हैं? कक्षाओं को थ्रेडेबल कैसे बनाया जाए; थ्रेड का विस्तार करना; रन
पाठ्यपुस्तक 2: अध्याय 11 8 घंटे
मॉड्यूल - V
ईवेंट हैंडलिंग: दो ईवेंट हैंडलिंग तंत्र; प्रतिनिधिमंडल ईवेंट मॉडल; ईवेंट कक्षाएँ; ईवेंट के स्रोत; ईवेंट लिसिनर इंटरफेस; प्रतिनिधिमं
स्विंग: स्विंग की उत्पत्ति; दो प्रमुख स्विंग विशेषताएँ; घटक और कंटेनर; स्विंग पैकेज; एक सरल स्विंग एप्लिकेशन
पाठ्यपुस्तक 2: अध्याय 22, 29 8 घंटे

पाठ्यक्रम परणिाम
इस पाठ्यक्रम के पूरा होने पर, छात्र नमि्नलखिति करने में सक्षम होंगे:
CO
संख्या।
पाठ्यक्रम परणिाम ववरिण ब्लूम की
वर्गीकरण
स्तर
CO1 वंशानुक्रम, बहुरूपता, नेस्टेड कक्षाएँ, कंस्ट्रक्टर, वनिाशक जैसे वभिन्नि ऑब्जेक्ट ओरएिंटेड अवधारणाओं का उपयोग व
CO2 डेटा प्रकारों, चरों और ऐरे, ऑपरेटरों, नयिंत्रण कथनों की सहायता से जावा के साथ बुनयिादी ऑब्जेक्ट ओरएिंटेड अवधारण
CO3 जावा का उपयोग करके वंशानुक्रम, अपवाद, पैकेज अवधारणाओं और अपवाद हैंडलगि का नरीक्षण करें। CL3
CO4 वास्तवकि समय के अनुप्रयोगों में थ्रेडगि और बहु-थ्रेड प्रोग्रामगि की अवधारणा का उपयोग करें। CL3
CO5 जावा ईवेंट हैंडलगि तंत्र को चत्रिति करें CL3
प्रयोगशाला घटक

अनुभव।
संख्या।
प्रयोग ववरिण CO
संख्या। ब्लूम की
वर्गीकरण
स्तर

1. इंजीनयिरगि कॉलेज से संबंधति छात्रों की बुनयिादी जानकारी रखने वाली कक्षा बनाकर कक्षा, ऑब्जेक्ट और कंस्ट्रक्टर क
CO1
CL3

2. पूर्णांक और फ्लोटगि पॉइंट संख्याओं पर बुनयिादी अंकगणतिीय संक्रयिाएँ करने वाले कैलकुलेटर का अनुकरण करने के लिए प
CO1
CL3

3. जावा कक्षाओं, ऑब्जेक्ट्स, कंस्ट्रक्टर, चरों की घोषणा और इनशियिलाइजेशन के नरि्माण को प्रदर्शति करने के लिए एक
CO2
CL3
4. जावा प्रोग्राम का उपयोग करके for, for-each, while और do-while लूप का प्रदर्शन करें CO2 CL3
5. वंशानुक्रम, बहुरूपता की अवधारणा को चत्रिति करने के लिए एक जावा प्रोग्राम लागू करें CO3 CL3 पृष्ठ 3 |
6. जावा पैकेज बनाने और उपयोगकर्ता द्वारा परभिाषति जावा पैकेज को आयात करने की प्रक्रयिा को चत्रिति करने के लिए एव
CO3
CL3

7. जावा मल्टी-थ्रेडगि अवधारणाओं का उपयोग करके बाउंडेड बफ़र समस्याओं के प्रदर्शन को प्रदर्शति करने के लिए एक जाव
CO4
CL3
8. उत्पादक-उपभोक्ता समस्याओं के प्रदर्शन को प्रदर्शति करने के लिए एक जावा प्रोग्राम लागू करें CO4 CL3
9. प्रमुख ईवेंट और माउस ईवेंट का अनुकरण करने के लिए एक जावा प्रोग्राम वकिसति करें CO5 CL3
10. जावा स्वगि को प्रदर्शति करने के लिए एक जावा प्रोग्राम वकिसति करें CO5 CL3
CO-PO-PSO मैपगि

CO
संख्या।

कार्यक्रम परिणाम (PO) कार्यक्रम
विशिष्ट
परिणाम (PSO)
1 2 3 4 5 6 7 8 9 10 11 12 1 2
CO1
CO2
CO3
CO4
CO5
3: पर्याप्त (उच्च) 2: मध्यम (मध्यम) 1: खराब (कम)

मूल्यांकन रणनीति

मूल्यांकन CIA और SEE दोनों होंगे। छात्रों के सीखने का मूल्यांकन प्रत्यक्ष और अप्रत्यक्ष विधियों का उपयोग करके किया ज
क्रं. सं. मूल्यांकन विवरण भारांक (%) अधिकतम अंक
1 सतत आंतरिक मूल्यांकन (CIA) 100% 50
सतत आंतरिक मूल्यांकन (CIE) 60% 30
व्यावहारिक सत्र (प्रयोगशाला घटक) 40% 20
2 सेमेस्टर अंत परीक्षा (SEE) 100% 50
मूल्यांकन विवरण
सतत आंतरिक मूल्यांकन (CIA) (50%) सेमेस्टर अंत परीक्षा (SEE) (50%)
सतत आंतरिक मूल्यांकन (CIE) (60%) व्यावहारिक सत्र (40%)
मैं द्वितीय तृतीय
पाठ्यक्रम कवरेज पाठ्यक्रम कवरेज पाठ्यक्रम कवरेज
40% 30% 30% 100% 100%
MI MI MI पृष्ठ 4 |
MII MII MII MII
MIII MIII MIII
MIV MIV MIV
MV MV MV

नोट :

 मूल्यांकन CIA और SEE दोनों होंगे।
 IPCC के व्यावहारिक सत्र केवल CIE के लिए होंगे।
 IPCC का सद्धिांत घटक क्रमशः CIA और SEE दोनों के लिए होगा।
 व्यावहारिक सत्रों से प्रश्न सद्धिांत SEE में शामिल किए जाएँगे।
नोट: परीक्षाओं (CIE और SEE दोनों) के लिए, प्रश्न पत्रों में उपयुक्त ब्लूम के स्तर से संबंधित प्रश्न होंगे। उच्च संज्ञानात्म
SEE प्रश्न पत्र पैटर्न:

1. प्रश्न पत्र में पाँच मॉड्यूल से दस पूर्ण प्रश्न होंगे
2. प्रत्येक मॉड्यूल से 2 पूर्ण प्रश्न होंगे। प्रत्येक प्रश्न अधिकतम 20 अंक का होगा।
3. प्रत्येक पूर्ण प्रश्न में अधिकतम चार उप-प्रश्न हो सकते हैं जो एक मॉड्यूल के अंतर्गत सभी विषयों को कवर करते हैं।
4. छात्रों को प्रत्येक मॉड्यूल से एक पूर्ण प्रश्न चुनकर पाँच पूर्ण प्रश्नों के उत्तर देने होंगे।
पाठ्य पुस्तकें:

1. हर्बर्ट शिल्ड्ट, जावा द कम्प्लीट रेफरेंस, 7वाँ संस्करण, टाटा मैकग्रा हिल, 2007।
संदर्भ पुस्तकें:

1. सौरव सहय, ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग विद C++, दूसरा संस्करण, ऑक्सफोर्ड यूनिवर्सिटी प्रेस, 2006
2. ई बालागुरुस्वामी, प्रोग्रामिंग विद जावा, मैकग्रा हिल, 6वाँ संस्करण, 2019।
3. महेश भावे और सुनील पाटेकर, "प्रोग्रामिंग विद जावा", प्रथम संस्करण, पियर्सन एजुकेशन, 2008,

ISBN:978813 1720806
संदर्भ वेब लिंक और वीडियो व्याख्यान (ई-संसाधन):

1. https://onlinecourses.nptel.ac.in/noc22_cs102/
2. https://www.geeksforgeeks.org

मॉड्यूल 1: ऑब्जेक्ट ओरिएंटेड कॉन्सेप्ट्स का परिचय

पाठ्यक्रम:

ऑब्जेक्ट ओरिएंटेड कॉन्सेप्ट्स का परिचय: संरचनाओं की समीक्षा, प्रक्रिया-उन्मुख प्रोग्रामिंग प्रणाली, ऑब्जेक्ट ओरिएंटे

कक्षा और ऑब्जेक्ट: परिचय, सदस्य फ़ंक्शन और डेटा, ऑब्जेक्ट और फ़ंक्शन। ऑब्जेक्ट और ऐरे, नामस्थान, नेस्टेड कक्ष

ऑब्जेक्ट ओरिएंटेड कॉन्सेप्ट्स का परिचय

संरचनाओं की समीक्षा

 संरचनाओं की आवश्यकता
समस्या:
• ऐसे मामले हो सकते हैं (चरों के समूहों का उपयोग करते समय) जहाँ एक चर का मान तार्किक रूप से अन्य चर के मान को प्रभाव
• हालाँकि, कोई भाषा निर्माण नहीं है जो वास्तव में इन चरों को एक ही समूह में रखता है। इस प्रकार, गलत समूह के सदस्यों को
• ऐरे का उपयोग इस समस्या को हल करने के लिए किया जा सकता है लेकिन यह काम नहीं करेगा यदि चर एक ही प्रकार के नहीं
समाधान: संरचनाओं का उपयोग करके एक डेटा प्रकार स्वयं बनाएँ।

```
struct date             // तिथियों का प्रतिनिधित्व करने के लिए एक संरचना
{
int d,m,y;
}

void next_day(struct date *);
```

एक संरचना सी में एक प्रोग्रामिंग निर्माण है जो हमें चरों को एक साथ रखने की अनुमति देता है जो एक साथ होने चाहिए।
• पुस्तकालय प्रोग्रामर नए डेटा प्रकार बनाने के लिए संरचनाओं का उपयोग करते हैं।
• अनुप्रयोग प्रोग्राम और अन्य पुस्तकालय प्रोग्राम इस डेटा प्रकार के चर घोषित करके इन नए डेटा प्रकारों का उपयोग करते

 संरचना का उपयोग करके नया डेटा प्रकार बनाना
यह एक तीन-चरणीय प्रक्रिया है जिसे लाइब्रेरी प्रोग्रामर द्वारा इस प्रकार निष्पादित किया जाता है:
चरण 1 संरचना की परिभाषा और संबद्ध कार्यों के प्रोटोटाइप को एक शीर्षलेख फ़ाइल में रखें। एक शीर्षलेख फ़ाइल में एक संर
चरण 2 संबद्ध कार्यों की परिभाषा को एक स्रोत कोड में रखें और एक लाइब्रेरी बनाएँ।
चरण 3 अन्य प्रोग्रामरों को हेडर फ़ाइल और लाइब्रेरी प्रदान करें, जो भी माध्यम हो, जो इस नए डेटा प्रकार का उपयोग करना
नोट: एक संरचना का निर्माण और इसके संबद्ध कार्यों का निर्माण दो अलग-अलग चरण हैं जो मिलकर एक पूरी प्रक्रिया बनाते

 अनुप्रयोग प्रोग्राम में संरचनाओं का उपयोग करना
चरण 1 स्रोत कोड में लाइब्रेरी प्रोग्रामर द्वारा प्रदान की गई शीर्षलेख फ़ाइल शामिल करें।
चरण 2 स्रोत कोड में नए डेटा प्रकार के चर घोषित करें।
चरण 3 स्रोत कोड में इन चरों को पास करके संबद्ध कार्यों के कॉल एम्बेड करें।

चरण 4 ऑब्जेक्ट फ़ाइल प्राप्त करने के लिए स्रोत कोड संकलित करें।
चरण 5 निष्पादन योग्य फ़ाइल या किसी अन्य लाइब्रेरी को प्राप्त करने के लिए लाइब्रेरी प्रोग्रामर द्वारा प्रदान की गई लाइब्रे

C++ का अवलोकन

 C++ एक्सटेंशन का आविष्कार पहली बार "ब्जार्ने स्ट्रौस्ट्रुप" ने 1979 में किया था।

 उन्होंने शुरू में नई भाषा को "सी विद क्लासेस" कहा था।

 हालाँकि1983 में नाम बदलकर C++ कर दिया गया।

 c++ C भाषा का एक विस्तार है, जिसमें अधिकांश C प्रोग्राम c++ प्रोग्राम भी हैं।

 C++, C के विपरीत, "ऑब्जेक्ट-ओरिएंटेड प्रोग्रामिंग" का समर्थन करता है।

ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग सिस्टम (OOPS)

 OOPS में हम वास्तविक दुनिया की वस्तुओं को मॉडल करने का प्रयास करते हैं।

 अधिकांश वास्तविक दुनिया की वस्तुओं में आंतरिक भाग (डेटा सदस्य) और इंटरफेस होते हैं
(सदस्य फ़ंक्शन) जो हमें उन्हें संचालित करने में सक्षम बनाते हैं।

ऑब्जेक्ट:

 दुनिया में सब कुछ एक ऑब्जेक्ट है।

 एक ऑब्जेक्ट चरों का एक संग्रह है जो डेटा रखते हैं और फ़ंक्शन जो डेटा पर काम करते हैं।
 डेटा रखने वाले चरों को डेटा सदस्य कहा जाता है।

 डेटा पर काम करने वाले कार्यों को सदस्य फ़ंक्शन कहा जाता है।

एक ऑब्जेक्ट के दो भाग:

 ऑब्जेक्ट = डेटा + विधियाँ (फ़ंक्शन)

 ऑब्जेक्ट ओरिएंटेड प्रोग्रामिंग में ध्यान एक कार्य को पूरा करने के लिए ऑब्जेक्ट बनाना है और प्रक्रियाएँ (फ़ंक्शन) नहीं ब
 OOPs में डेटा फ़ंक्शन से अधिक निकटता से जुड़ा होता है और डेटा को पूरे प्रोग्राम में स्वतंत्र रूप से प्रवाहित नहीं होने देता है
 डेटा छिपा हुआ है और बाहरी कार्यों द्वारा आसानी से पहुँचा नहीं जा सकता है। पृष्ठ 8 |

 OOP को लागू करने वाले कंपाइलर अनधिकृत कार्यों को डेटा तक पहुँचने की अनुमति नहीं देते हैं, जिससे डेटा सुरक्षा सुनिश्चि

 केवल संबद्ध फ़ंक्शन ही डेटा पर काम कर सकते हैं और प्रोग्राम में बग आने की कोई संभावना नहीं है।
 OOP का मुख्य लाभ वास्तविक दुनिया की समस्याओं को मॉडल करने की इसकी क्षमता है।

 यह प्रोग्राम डिज़ाइन में बॉटम-अप दृष्टिकोण का पालन करता है।
ऑब्जेक्ट A ऑब्जेक्ट B ऑब्जेक्ट C

डेटा डेटा डेटा

फ़ंक्शन फ़ंक्शन फ़ंक्शन

संचार

 वस्तुओं की पहचान करना और इन वस्तुओं को जम्मिेदारयिाँ सौंपना।

 ऑब्जेक्ट संदेश भेजकर अन्य ऑब्जेक्ट्स से संवाद करते हैं।
 संदेश एक ऑब्जेक्ट के तरीकों (फ़ंक्शन) द्वारा प्राप्त कएि जाते हैं।

ऑब्जेक्ट-ओरएिंटेड प्रोग्रामगि की बुनयिादी अवधारणाएँ (सुवधिाएँ)

1. ऑब्जेक्ट

2. कक्षाएँ

3. डेटा अमूर्तता

OOP के तीन स्तंभ:

4. डेटा इनकैप्सुलेशन

5. वंशानुक्रम

6. बहुरूपता

 ऑब्जेक्ट और कक्षाएँ:

 कक्षाएँ उपयोगकर्ता द्वारा परभिाषति डेटा प्रकार हैं जनि पर ऑब्जेक्ट बनाए जाते हैं।

 समान गुणों और वधियिों वाले ऑब्जेक्ट्स को एक साथ समूहीकृत करके कक्षा बनाई जाती है।
 तो कक्षा ऑब्जेक्ट का एक संग्रह है।

 ऑब्जेक्ट एक कक्षा का उदाहरण है।

 डेटा अमूर्तता

 अमूर्तता आवश्यक सुवधिाओं का प्रतनिधित्वि करने के कार्य को संदर्भति करती है जसिमें पृष्ठभूमि विविरण या स्पष्टीकरण
 उदाहरण: आइए एक वास्तवकि जीवन का उदाहरण लें टीवी का, जसिे आप चालू और बंद कर सकते हैं, चैनल बदल सकते हैं, वॉ
 उदाहरण: #include <iostream>

```
int main( )
{
cout << "Hello C++" <<endl;
return 0;
}
```

 यहाँ, आपको यह समझने की आवश्यकता नहीं है कि cout उपयोगकर्ता की स्क्रीन पर पाठ कैसे प्रदर्शति करता है। आपको
 डेटा इनकैप्सुलेशन

 सूचना छपिाना

 डेटा और कार्यों को एकल इकाई (कक्षा) में लपेटना (संयोजन) को डेटा इनकैप्सुलेशन के रूप में जाना जाता है।
 डेटा बाहरी दुनिया के लिए सुलभ नहीं है, केवल वे फ़ंक्शन जो कक्षा में लिपटे हुए हैं वे इसे एक्सेस कर सकते हैं।
 वंशानुक्रम

 गुण प्राप्त करना।

 मौजूदा कक्षा से नई कक्षा प्राप्त करने की प्रक्रिया।

 मौजूदा कक्षा को बेस, पैरेंट या सुपर क्लास के रूप में जाना जाता है।

 बनने वाली नई कक्षा को व्युत्पन्न कक्षा, चाइल्ड या सब क्लास कहा जाता है।

 व्युत्पन्न कक्षा में बेस कक्षा की सभी विशेषताएँ होती हैं साथ ही इसमें कुछ अतिरिक्त विशेषताएँ भी होती हैं।
 पुन: प्रयोज्य कोड लिखना।

 ऑब्जेक्ट अन्य ऑब्जेक्ट्स से विशेषताएँ प्राप्त कर सकते हैं। पृष्ठ 9 |

 बहुरूपता

 बहुरूपता का शब्दकोश अर्थ है "बहु रूप"।

 एक से अधिक रूप लेने की क्षमता।

 एक ही नाम के कई अर्थ हो सकते हैं जो इसके संदर्भ पर निर्भर करता है।

 इसमें फ़ंक्शन अधिभारण, ऑपरेटर अधिभारण शामिल है।

OOP में प्रोग्रामिंग की प्रक्रिया में निम्नलिखित बुनियादी चरण शामिल हैं:

1. ऑब्जेक्ट और व्यवहार को परिभाषित करने वाली कक्षाएँ बनाना।

2. कक्षा परिभाषाओं से ऑब्जेक्ट बनाना।

3. ऑब्जेक्ट के बीच संचार स्थापित करना।

OOP के लाभ

 डेटा सुरक्षा

 मौजूदा कोड का पुन: उपयोग
 नए डेटा प्रकार बनाना
 अमूर्तता
 कम विकास समय
 जटिलता कम करें
 बेहतर उत्पादकता

OOP के लाभ

 पुन: प्रयोज्यता

 वकिास समय की बचत और उच्च उत्पादकता
 डेटा छपिाना
 कई ऑब्जेक्ट सुवधिा
 ऑब्जेक्ट के आधार पर प्रोजेक्ट में काम को वभिाजति करना आसान है।
 छोटे से बड़े ससि्टम में उन्नयन
 इंटरफ़ेस के लएि संदेश पासगि तकनीक।
 सॉफ़्टवेयर जटलिता को आसानी से प्रबंधति कयिा जा सकता है।

OOP के अनुप्रयोग

 वास्तवकि समय प्रणाली पृष्ठ 10 |

 समिुलेशन और मॉडलगि
 ऑब्जेक्ट ओरएिंटेड डेटाबेस
 हाइपरटेक्स्ट, हाइपरमीडयिा
 एआई (कृत्रमि बुद्धमित्ता)
 तंत्रकिा नेटवर्क और समानांतर प्रोग्रामगि
 नर्िणय समर्थन और कार्यालय स्वचालन प्रणाली
 CIM/CAD/CAED प्रणाली

POP(प्रक्रयिा उन्मुख प्रोग्रामगि) और OOP(ऑब्जेक्ट ओरएिंटेड प्रोग्रामगि) के बीच अंतर

क्रं.सं POP OOP
1. प्रक्रयिाओं (फ़ंक्शन) पर ज़ोर डेटा पर ज़ोर
2. प्रोग्रामगि कार्य डेटा संरचनाओं और फ़ंक्शनों के संग्रह में वभिाजति है। प्रोग्रामगि कार्य ऑब्जेक्ट्स में वभिाजति है
(डेटा चर और संबद्ध सदस्य कार्यों से मलिकर)
3. प्रक्रयिाओं को हेरफेर कएि जा रहे डेटा से अलग कयिा जा रहा है प्रक्रयिाएँ डेटा से अलग नहीं हैं,
बल्कि प्रक्रयिाएँ और डेटा एक साथ संयुक्त हैं।
4. एक कोड वशिष्टि कार्य करने के लएि डेटा का उपयोग करता है डेटा वशिष्टि कार्य करने के लएि कोड का उपयोग करता है
5. डेटा को पैरामीटर का उपयोग करके एक फ़ंक्शन से दूसरे फ़ंक्शन में स्वतंत्र रूप से स्थानांतरति कयिा जाता है। डेटा छपिा हुआ है
6. डेटा सुरक्षति नहीं है डेटा सुरक्षति है
7. प्रोग्राम डज़िाइन में टॉप-डाउन दृष्टकिोण का उपयोग कयिा जाता है प्रोग्राम डज़िाइन में बॉटम-अप दृष्टकिोण का उपयोग कि
8. डीबगगि कठनि है क्योंकि कोड का आकार बढ़ता है डीबगगि आसान है, भले ही कोड का आकार अधकि हो पृष्ठ 11 |

C और C++ की तुलना

क्रं.सं C C++
1. यह प्रक्रयिा उन्मुख भाषा है यह ऑब्जेक्ट-ओरएिंटेड भाषा है
2. उन कार्यों को लखिने पर जोर दयिा जाता है जो कुछ वशिष्टि कार्य करते हैं। डेटा पर जोर दयिा जाता है जो कार्य को प्राप्त क
3. डेटा और फ़ंक्शन अलग हैं डेटा और फ़ंक्शन संयुक्त हैं
4. बहुरूपता, वंशानुक्रम आदि का समर्थन नहीं करता है। बहुरूपता, वंशानुक्रम आदि का समर्थन करता है।
5. वे तेजी से चलते हैं समकक्ष C प्रोग्राम की तुलना में वे धीमी गति से चलते हैं
6. प्रकार जांच इतनी मजबूत नहीं है प्रकार जांच बहुत मजबूत है
7. लाखों लाइनों के कोड प्रबंधन में बहुत कठनिाई होती है लाखों लाइनों के कोड को बहुत आसानी से प्रबंधति कयिा जा सकता है
8. फ़ंक्शन परभिाषा और घोषणाओं को संरचना परभिाषाओं के भीतर अनुमति नहीं है फ़ंक्शन परभिाषा और घोषणाओं को संरचना

C++ में कंसोल आउटपुट/इनपुट

Cin: कीबोर्ड इनपुट के लिए उपयोग कयिा जाता है।

Cout: स्क्रीन आउटपुट के लिए उपयोग कयिा जाता है।

चूँकि Cin और Cout C++ ऑब्जेक्ट हैं, वे कुछ हद तक "बुद्धमिान" हैं।

 उन्हें सामान्य प्रारूप स्ट्रगि और रूपांतरण वनिर्दिेशों की आवश्यकता नहीं है।

 वे स्वचालति रूप से जानते हैं कि किसि प्रकार के डेटा शामलि हैं।

 उन्हें पता ऑपरेटर और की आवश्यकता नहीं है,

 उन्हें स्ट्रीम नष्किर्षण (>>) और सम्मलिन (<<) ऑपरेटरों के उपयोग की आवश्यकता है।

नष्किर्षण ऑपरेटर (>>):

 कीबोर्ड से इनपुट प्राप्त करने के लिए हम नष्किर्षण ऑपरेटर और ऑब्जेक्ट Cin का उपयोग करते हैं।
 सटिैक्स: Cin>> चर;

 चर के सामने "और" की कोई आवश्यकता नहीं है।

 कंपाइलर चर के प्रकार का पता लगाता है और उपयुक्त प्रकार में पढ़ता है। पृष्ठ 12 |

o उदाहरण:

```
#include<iostream.h>
Void main( )
{
int x;
float y;
cin>> x;
cin>>y;
}
```

सम्मलिन ऑपरेटर (<<):

 स्क्रीन पर आउटपुट भेजने के लिए हम ऑब्जेक्ट Cout पर सम्मलिन ऑपरेटर का उपयोग करते हैं।

 सटिैक्स: Cout<<चर;

 कंपाइलर ऑब्जेक्ट के प्रकार का पता लगाता है और इसे उचति रूप से प्रटि करता है।

उदाहरण:

```
#include<iostream.h>
void main( )
{



}
```

प्रोग्राम cout<<5;
cout<<4.1;
cout<< "string";
cout<< '\n';

Cin और Cout का उपयोग करके उदाहरण

```
#include<iostream.h>
void main( )
{
int a,b;
float k;
char name[30];
cout<< "अपना नाम दर्ज करें \n";
cin>>name;
cout<< "दो पूर्णांक और एक फ्लोट दर्ज करें \n";
cin>>a> >b>>k;
cout<< "धन्यवाद," <<name<<",आपने दर्ज कयिा है\n";
cout<<a<<","<<b<<",और"<<k<<'/n';
}
```

आउटपुट:
अपना नाम दर्ज करें : महेश पृष्ठ 13 |

दो पूर्णांक और एक फ्लोट दर्ज करें
10
20
30.5
धन्यवाद महेश, आपने दर्ज कयिा है
10, 20 और 30.5

C++ प्रोग्राम कसिी संख्या का वर्ग ज्ञात करने के लिए

```
#include<iostream.h>
int main( )
{
int i;
cout<< "यह आउटपुट है\n";
cout<< "एक संख्या दर्ज करें";
cin>>i;
cout<<i<< "वर्ग है" << i*i<< "\n";
return 0;
}
```

आउटपुट:
यह आउटपुट है
एक संख्या दर्ज करें 5

5 वर्ग है 25

चर

चरों का उपयोग C++ में कयिा जाता है, जहाँ हमें कसिी भी मान के लिए संग्रहण की आवश्यकता होती है, जो प्रोग्राम में बदल जा
चर को कई तरह से घोषति कयिा जा सकता है, प्रत्येक की अलग-अलग मेमोरी आवश्यकताएँ और कार्यप्रणाली होती है। चर कंप

घोषणा और आरंभीकरण

 चरों का उपयोग करने से पहले उन्हें घोषति कयिा जाना चाहिए। आमतौर पर उन्हें प्रोग्राम की शुरुआत में घोषति करना बेहतर हो

उदाहरण :

```
int i;       // घोषति कयिा गया लेकनि आरंभ नहीं कयिा गया
char c;
int i, j, k;  // एकाधकि घोषणा
```

आरंभीकरण का अर्थ है पहले से घोषति चर को मान नर्दिष्टि करना,

```
int i;  // घोषणा
i = 10;  // आरंभीकरण
```

आरंभीकरण और घोषणा एक ही चरण में भी की जा सकती है,

```
int i=10;

int i=10, j=11; //एक ही चरण में आरंभीकरण और घोषणा पृष्ठ 15 | \
```

 यदकिोई चर घोषति कयिा जाता है और आरंभ नहीं कयिा जाता है तो डफ़िॉल्ट रूप से यह एक गैरबेज मान रखेगा।
इसके अलावा, यदकिोई चर एक बार घोषति हो जाता है और यदहिम इसे फरि से घोषति करने का प्रयास करते हैं, तो हमें संकलन

```
int i,j;
i=10;
j=20;
int j=i+j;       //संकलन समय त्रुटि, एक ही स्कोप में चर को फरि से घोषति नहीं कर सकते
```

चर का दायरा

सभी चरों के कार्य करने का अपना क्षेत्र होता है, और उस सीमा से बाहर वे अपना मान नहीं रखते हैं, इस सीमा को चर का दायरा

• वैश्वकि चर

• स्थानीय चर

वैश्वकि चर

वैश्वकि चर वे होते हैं, जो एक बार घोषति कएि जाते हैं और प्रोग्राम के पूरे जीवनकाल में कसिी भी वर्ग या कसिी भी फ़ंक्शन द्वा