अध्याय 8

C++ में फंक्शन

8.1 परिचय

फंक्शन प्रोग्राम का एक भाग होता है जिसे किसी कार्य के लिए प्रयोग किया जाता है। एक प्रोग्राम का फंक्शनस में विभाजित करना प्रोग्रामिंग भाषा के मुख्य सिद्वान्तों में से एक है। प्रोग्राम में विभिन्न स्थानों पर कॉलिंग और उपयोग करके प्रोग्राम के आकार का कम करता है। C++में फंक्शन को और अधिक विश्वसनीय और लचीला बनाने के लिए कई नये फिचर जोड़े गये है।

8.2 फंक्शन प्रोटोटाईप

फंक्शन प्रोटोटाईप कम्पाइलर को फंक्शन के बारे में सूचना देता है। जैसे आरग्यूमेन्ट की संख्या और उनके टाईप और रिटर्न टाईप। फंक्शन प्रोटोटाईप एक कॉलिंग प्रोग्राम में घोषणा करने वाला स्टेटमेन्ट होता हैं और इसका सिन्टेक्स निम्न प्रकार का होता है।

```
type function_name(arguments-list);
```

उदाहरण :–

```
int sum(int a, int b);
```

फंक्शन की घोषणा में आरग्यूमेन्ट के नाम छदम वेरिएबल होते है और वे ऐच्छिक होते है।

```
int sum(int, int);
```

एक वैध फंक्शन की घोषणा है।

8.3 कॉल–बाई–रेफरेंस

कॉल–बाई–वेल्यू पेरामीटर भेजने वाली विधि में कॉल्ड फंक्शन कॉलिंग प्रोग्राम में वास्तविक पेरामीटर को कॉल्ड फंक्शन द्वारा फोरमल पेरामीटर में कॉपी किया जाता है। फोरमल पेरामीटर पर किये बदलाव कॉलिंग प्रोग्राम में नहीं होता है। कॉलिंग प्रोग्राम के वास्तविक पेरामीटर मे बदलाव के लिए हम कॉल–बाई–रेफरेंस पेरामीटर भेजने वाली विधि का प्रयोग करते हैं

उदाहरण :–

प्रोग्राम : 8.1 कॉल–बाई–रेफरेंस

```
#include<iostream>
using namespace std;
int  main()
{
```

```
int count=0;
void update(int &);
cout<<"count="<<count<<"\n";
update(count);
cout<<"count="<<count;
return 0;
}
void update(int &x)
{
        x=x+1;
}
```

प्रोग्राम 8.1 का आउटपुट होगा–

```
count=0
count=1
```

यहाँ पर कॉल्ड फंक्शन update में वेरिएबल x, main फंक्शन में वेरिएबल count का उपनाम बन जाता है। x की वेल्यू में एक की बढोतरी से main फंक्शन में count की वेल्यू बढ़ जाती है।

8.4 रिटर्न–बाई–रेफरेंस

एक फंक्शन रेफरेंस भी रिटर्न कर सकता है।

प्रोग्राम : 8.2 रिटर्न–बाई–रेफरेंस

```
#include<iostream>
using namespace std;
int main()
{
        int x=6,y=9;
        int & min(int &, int &);
        min(x,y)=-1;
        cout<<"x="<<x<<"\n";
```

```
        cout<<"y="<<y;
        return 0;
}

int & min(int &a, int &b)
{
          if(a<b)
          return a;
          else
          return b;
}
```

प्रोग्राम 8.2 का आउटपुट होगा–

x=-1

y=9

8.5 फंक्शन ओवरलोडिंग

फंक्शन के एक समान नाम लेकिन उनकी आरग्यूमेन्ट लिस्ट में भिन्नता और अलग–अलग कार्य करना इसे फंक्शन ओवरलोडिंग कहा जाता है। सही फंक्शन को कॉल करना उसके आरग्यूमेन्ट की संख्या और टाईप पर निर्भर करता है न की उसके रिटर्न टाईप पर।

प्रोग्राम : 8.3 फंक्शन ओवरलोडिंग

```
#include<iostream>
using namespace std;
int sum(int, int);
int sum(int, int, int);
int main()
{
        cout<<"Sum of two numbers is "<<sum(5,10);
        cout<<"\n";
        cout<<"Sum of three numbers is "<<sum(10,20,30);
        return 0;
```

```
}
int sum(int x, int y)
{
        return(x+y);
}
int sum(int a, int b, int c)
{
        return(a+b+c);
}
```

प्रोग्राम 8.3 का आउटपुट होगा–

```
Sum of two numbers is 15
Sum of three numbers is 60
```

उपरोक्त प्रोग्राम में sum() फंक्शन को ओवरलोड किया गया है। जब हम sum() फंक्शन को दो आरग्यूमेन्ट भेजते है तब दो आरग्यूमेन्ट वाला फंक्शन कॉल होता है। जब हम sum() फंक्शन को तीन आरग्यूमेन्ट भेजते है तब तीन आरग्यूमेन्ट वाला फंक्शन कॉल होता है।

8.6 इनलाईन

जब किसी फंक्शन को कॉल किया जाता हैं संपादन का कन्ट्रोल कॉलिंग फंक्शन से कॉल्ड फंक्शन को चला जाता हैं। कॉल्ड फंक्शन संपादन होने के बाद संपादन का कन्ट्रोल पुनः कॉलिंग फंक्शन को चला जाता हैं। जब फंक्शन का आकार छोटा होता है तब इस प्रकिया में काफी समय बर्बाद हो जाता है। इस समस्या का समाधान इनलाईन फंक्शन है। एक इनलाईन फंक्शन का विस्तार इसे कॉल करने पर होता है। कम्पाइलर फंक्शन कॉल स्टेटमेंट को उसकी बॉडी से विस्थापित कर देता है।

महत्वपूर्ण बिंदु

- एक फंक्शन प्रोग्राम का भाग होता है जिसे किसी कार्य के लिए प्रयोग किया जाता है।
- फंक्शन प्रोटोटाईप कम्पाइलर को फंक्शन के बारे में सूचना देता है।
- एक फंक्शन रेफरेंस भी रिटर्न कर सकता है।
- फंक्शन के एक समान नाम लेकिन उनकी आरग्यूमेन्ट लिस्ट में भिन्नता और अलग–अलग कार्य करना इसे फंक्शन ओवरलोडिंग कहा जाता है।
- एक इनलाईन फंक्शन का विस्तार इसे कॉल करने पर होता है जिसमें कम्पाइलर फंक्शन कॉल स्टेटमेंट को उसकी बॉडी से विस्थापित कर देता है।

अभ्यासार्थ प्रश्न

वस्तुनिष्ठ प्रश्न :

प्रश्न 1. एक वैध फंक्शन की घोषणा कौनसी है?

(अ) int fun(int a, int b); (ब) int fun(int, int);

(स) अ. और ब. दोनों (द) इनमें से कोई नहीं

प्रश्न 2. पेरामीटर भेजने वाली किस विधि में कॉलिंग प्रोग्राम में वास्तविक पेरामीटर को कॉल्ड फंक्शन के फोरमल पेरामीटर में कॉपी किया जाता है

(अ) कॉल–बाई–रेफरेंस (ब) कॉल–बाई–वेल्यू

(स) कॉल–बाई–एड्रेस (द)) इनमें से कोई नहीं

प्रश्न 3. फंक्शन ओवरलोडिंग में सही फंक्शन को कॉल करना किस पर निर्भर नहीं करता है?

(अ) आरग्यूमेन्ट की संख्या (ब) आरग्यूमेन्ट के टाईप

(स) फंक्शन केरिटर्न टाईप (द) इनमें से कोई नहीं

प्रश्न 4. एक समान फंक्शन के नाम लेकिन उनकी आरग्यूमेन्ट लिस्ट में भिन्नता और अलग–अलग कार्य कर सकते है इसे कहा जाता है

(अ) फंक्शन ओवरलोडिंग (ब) ऑपरेटर ओवरलोडिंग

(स) क्लास ओवरलोडिंग (द) इनमें से कोई नहीं

अति लघूउत्तरात्मक प्रश्न

प्रश्न 1. फंक्शन किसे कहते है ?

प्रश्न 2. ऑपरेटर ओवरलोडिंग किसे कहते है ?

प्रश्न 3. इनलाईन फंक्शन किसे कहते है ?

लघूउत्तरात्मक प्रश्न

प्रश्न 1. फंक्शन प्रोटोटाईप किसे कहते है ?

प्रश्न 2. कॉल –बाई– वेल्यू और कॉल –बाई– रेफरेंस में क्या अन्तर है ?

प्रश्न 3. स्ट्रक्चरड प्रोग्रामिंग में फंक्शन के क्या फायदे है ?

निबंधात्मक प्रश्न

प्रश्न 1. दो वेल्यूज को आपस में अदला–बदली कॉल–बाई–रेफरेंस प्रणाली के द्वारा करने का प्रोग्राम लिखे ?

प्रश्न 2. वृत्त के क्षेत्रफल और आयत के क्षेत्रफल की गणना के लिए फंक्शन को ओवरलोड करने का प्रोग्राम लिखे ?

उत्तरमाला

1: स 2: ब 3: स 4: अ