

C++ में फंक्शन

पाठ्यपुस्तक के 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

उत्तर: (स) अ. और ब. दोनों

प्रश्न 2. पैरामीटर भेजने वाली किस विधि में कॉलिंग प्रोग्राम में वास्तविक पैरामीटर को कॉल्ड फंक्शन के फॉर्मल पैरामीटर में कॉपी किया जाता है?

- (अ) कॉल-बाई-रेफरेंस
- (ब) कॉल-बाई-वैल्यू
- (स) कॉल-बाई-एड्रेस ।
- (द) इनमें से कोई नहीं

उत्तर: (ब) कॉल-बाई-वैल्यू

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 के रिटर्न टाईप
- (द) इनमें से कोई नहीं

उत्तर: (स) फंक्शन के रिटर्न टाईप

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

- (अ) फंक्शन ओवरलोडिंग
- (ब) ऑपरेटर ओवरलोडिंग
- (स) क्लास ओवरलोडिंग
- (द) इनमें से कोई नहीं

उत्तर: (अ) फंक्शन ओवरलोडिंग

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. फंक्शन किसे कहते हैं?

उत्तर- फंक्शन प्रोग्राम का एक भाग होता है जिसे किसी कार्य के लिए प्रयोग किया जाता है। एक प्रोग्राम को फंक्शन में विभाजित करना प्रोग्रामिंग भाषा के मुख्य सिद्धान्तों में से एक है।

प्रश्न 2. फंक्शन ओवरलोडिंग किसे कहते हैं?

उत्तर- फंक्शन के एक समान नाम लेकिन उनकी आरग्यूमेन्ट लिस्ट में भिन्नता और अलग-अलग कार्य करना इसे फंक्शन ओवरलोडिंग कहा जाता है।

प्रश्न 3. इनलाईन फंक्शन किसे कहते हैं?

उत्तर- इनलाईन फंक्शन-जब किसी फंक्शन को कॉल किया जाता है तो संपादन का कन्ट्रोल कॉलिंग फंक्शन से कॉल्ड फंक्शन को चला जाता है। कॉल्ड फंक्शन संपादन होने के बाद संपादन का कन्ट्रोल पुनः कॉलिंग फंक्शन को चला जाता है। जब फंक्शन का आकार छोटा होता है। तब इस प्रक्रिया 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 स्टेटमेन्ट को उसकी बॉडी से विस्थापित कर देता है।

लघु उत्तरीय प्रश्न

प्रश्न 1. फंक्शन प्रोटोटाइप किसे कहते हैं?

उत्तर- फंक्शन प्रोटोटाइप-फंक्शन प्रोटोटाइप कम्पाइलर को फंक्शन के बारे में सूचना देता है। जैसे आरग्यूमेन्ट की संख्या और उनके टाईप और रिटर्न टाईप । फंक्शन प्रोटोटाइप 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 हैं और वे ऐच्छिक होते हैं।

`int sum(int, int);`

एक वैध फंक्शन की घोषणा है।

प्रश्न 2. कॉल-बाई-वैल्यू और कॉल-बाई-रेफरेंस में क्या अन्तर है?

उत्तर- कॉल-बाई-वेल्यू-कॉल-बाई-वेल्यू पैरामीटर भेजने वाली विधि में कॉल्ड फंक्शन कॉलिंग प्रोग्राम में वास्तविक पैरामीटर को कॉल्ड फंक्शन द्वारा फॉर्मल पैरामीटर में कॉपी किया जाता है। फॉर्मल पैरामीटर पर किये बदलाव कॉलिंग प्रोग्राम में नहीं होता है।

कॉल-बाई-रेफरेंस-जब रेफरेंस (reference) के द्वारा फंक्शन में पैरामीटर वेरिएबलस declare करते हैं तो ये वेरिएबल भी उसी memory location को point करते हैं जिसे original वेरिएबल point करते हैं। तो जब इन वेरिएबलस पर कोई चेंज perform करते हैं तो original वेरिएबलस भी चेंज हो जाते हैं।

प्रश्न 3. स्ट्रक्चर्ड प्रोग्रामिंग में फंक्शन के क्या फायदे हैं?

उत्तर- स्ट्रक्चर्ड प्रोग्रामिंग में फंक्शन के निम्नलिखित फायदे हैं

- Function create करने से प्रोग्रामर का समय और कम्प्यूटर की मैमोरी दोनों बचते हैं।
- एक ही code को आसानी से बार-बार use करने से re-usability बढ़ती है।
- प्रोग्राम फंक्शन में divide हो जाता है जिससे उसे आसानी से manage और debug किया जा सकता है।
- प्रोग्राम की readability बढ़ती है।

निबंधात्मक प्रश्न

प्रश्न 1. दो वेल्यूज को आपस में अदला-बदली कॉल-बाई-रेफरेंस प्रणाली के द्वारा करने का प्रोग्राम लिखें।

उत्तर:

```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
int a, b;
void swap(int &, int&);
cout<<"Enter two values:";
cin>>a>>b;
cout<<"\n Before sqapping: a =" <<a<<"b ="<<b;
swap (a, b);
cout << "\n After swapping: a=" << a<< "b =" <<b;
getch();
}
```

```

void swap (int & x,int &y)
{
int temp;
temp = x;
x = y;
y = temp;
}

```

Output

Enter two values : 5 10

Before swapping a = 5 b= 10

After swapping a = 10 b= 5

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

उत्तर- प्रोग्राम

```

#include<iostream.h>
#include<conio.h>
class over
{
float l, b, r, ar;
public :
void area (float, float);
void area (float);
};
void over :: area (float l, float b)
{
cout <<"Area of rectangle ="<< l *b;
}
void over :: area (float r)
{
cout<<"Area of circle ="<< 3.14 *r *r ;
}
void main()
{
over o;
clr sec();
float r, l, b;

```

```
cout<<"\n Enter radius";
cin>> r;
O. area (r);
cout<<"\n \n Enter length and breadth :";
cin>> l>>b;
O.area (l, b) ;
getch();
}
```

Output

Enter radius: 5

Area of circle = 78.5

Enter length and breadth: 3 4

Area of rectangle = 12

अन्य महत्वपूर्ण प्रश्न

अतिलघु उत्तरीय प्रश्न

प्रश्न 1. फंक्शन क्या होता है?

उत्तर- फंक्शन प्रोग्राम का एक भाग होता है जिसे किसी कार्य के लिए प्रयोग किया जाता है।

प्रश्न 2. फंक्शन प्रोटोटाइप का सिन्टेक्स बताइए।

उत्तर- सिन्टेक्स (Syntax)-type function_name (argument-list)

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

उत्तर- सही फंक्शन को कॉल करना उसके आरग्यूमेन्ट की संख्या और टाईप पर निर्भर करता है।

प्रश्न 4. जब किसी फंक्शन को कॉल किया जाता है तो संपादन का कन्ट्रोल कहाँ जाता है?

उत्तर- जब किसी फंक्शन को कॉल किया जाता है तो संपादन का कन्ट्रोल कॉलिंग फंक्शन से कॉल्ड फंक्शन को चला। जाता है।

लघु उत्तरीय प्रश्न

प्रश्न 1. कॉल-बाई-रेफरेंस का प्रोग्राम लिखिए।

अथवा

कॉल-बाई-रेफरेंस का उदाहरण दीजिए।

उत्तर- प्रोग्राम 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 का आउटपुट होगा
count=0
count=1
```

प्रश्न 2. रिटर्न-बाई-रेफरेंस को एक प्रोग्राम सहित समझाइए।

उत्तर- रिटर्न-बाई-रेफरेंस-एक फंक्शन रेफरेंस भी रिटर्न कर सकता है।
प्रोग्राम: 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 का आउटपुट होगा

```

x=-1
y= 9

```

निबंधात्मक प्रश्न

प्रश्न 1. फंक्शन ओवरलोडिंग का एक प्रोग्राम लिखिए।

अथवा

फंक्शन ओवरलोडिंग द्वारा दो व तीन नम्बरों को जोड़ने का प्रोग्राम लिखिए।

उत्तर- प्रोग्राम 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 का आउटपुट होगा

Sum of two numbers is 15

Sum of three numbers is 60